6 Eylül 2012 Rixos Pera İstanbul

Impact2012

Comes To You

Java and WAS on System z : Flexible, Fast, and Economic

First and Foremost … the Mainframe is NOT Dead!

"I predict that the last mainframe will be unplugged on March 15, 1996."

- Stewart Alsop, former InfoWorld columnist (now at Fortune Magazine), March, 1991

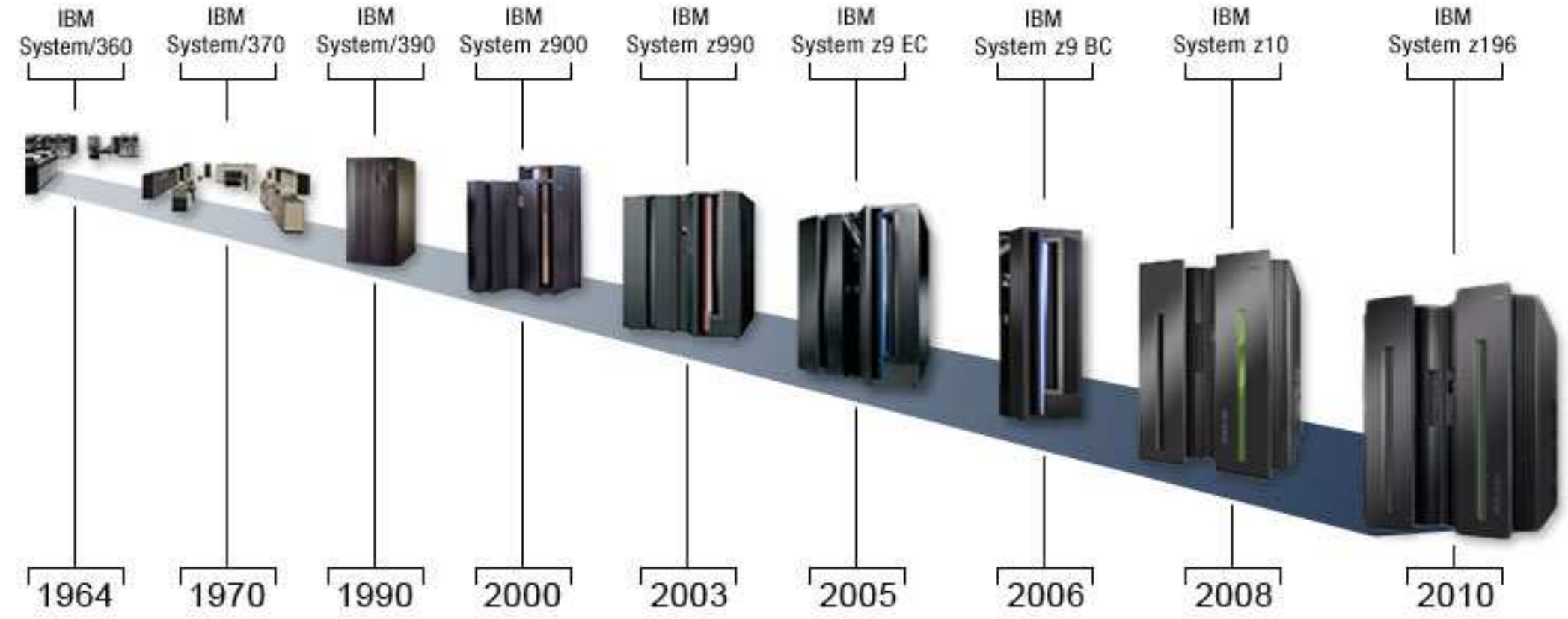
System z

- System z remains at the IT center of many large enterprises, its not just another computer system, its special.
- It holds the system of record for many large banks, insurance companies, and other industries. Why?
  - History and track record, it is there, it has done the job for years
  - It stays up, availability is critical, the Z in System z is for Zero Downtime. Systems have stayed up for decades.
  - The concept of availability is taken to the next level with things like geo-plex and the focus on disaster recovery
  - Scale, the z/OS Workload Manager manages varying workloads to high degrees of system utilization, and automatically manages to service level policies.
  - Security, the z/OS operating system is known for security and system integrity, specialized hardware is in place to separate the kernel operating system from the applications, and from differing pieces of the OS
- System z, where ***Businesses come to excel***

Let’s discuss some myths about System z ...

Myth 1 : IBM isn't investing in the Mainframe anymore!

False!



Impact2012 Comes To You

Now, announced August 28 2012, the new zEnterprise EC12 !

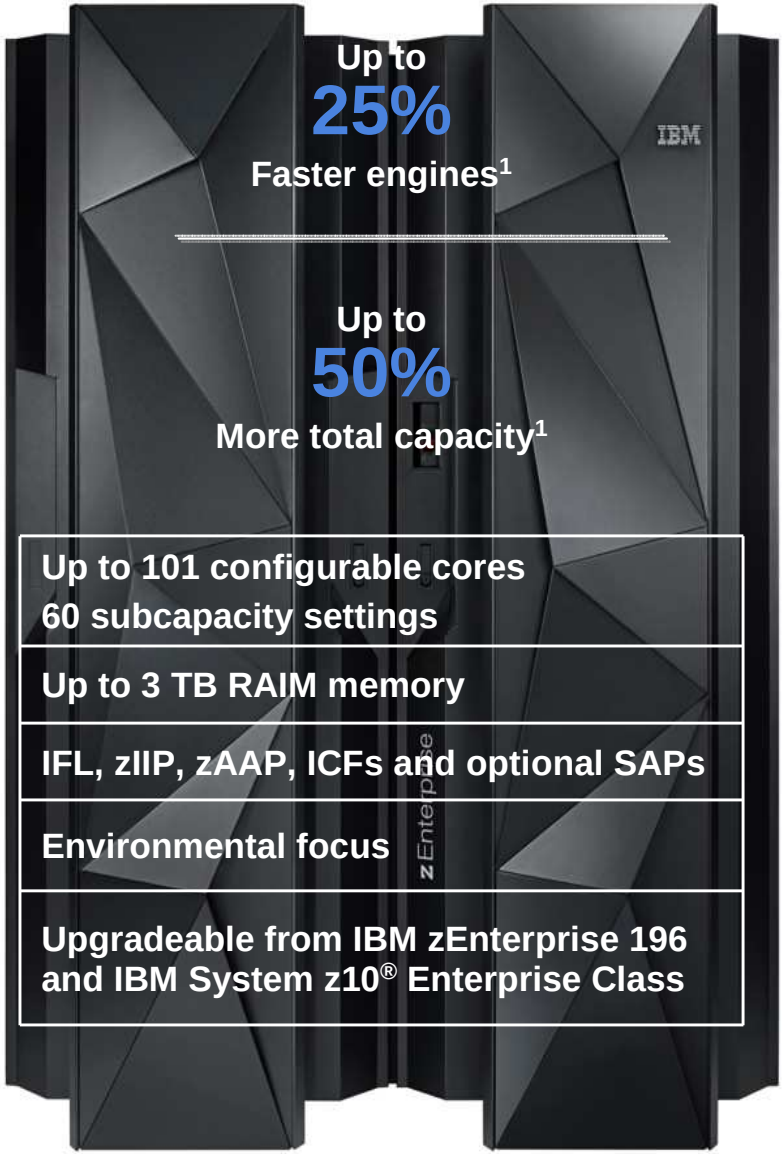
zEnterprise EC12 is the core of next generation System z

zEC12

Machine Type: 2827

- *Models: H20, H43, H66, H89, HA1*

- **Advanced Technology 5.5 GHz processor chip for performance boost for all workloads**
  - Over ***78,000 MIPS*** for large scale consolidation
  - ***Larger cache*** for data serving
- **Processor chip optimized for software performance – exploited by *Java, PL/I, compilers, DB2*** and more
- **Innovation to drive availability to superior levels**
  - ***IBM zAware*** offers snap-shot of the current state of your business
  - ***FLASH Express and pageable large pages*** to drive availability and performance for critical workloads
- **Security and reliability are in our DNA**
  - High speed ***cryptography integrated as part of the chip***
  - Enhanced functions with new ***Crypto Express4S***
  - PR/SM designed for ***EAL5+ certification***



Impact2012
Comes To You

Myth 2 : System z is too expensive to deploy today's Workloads!

False!

	Intel	Intel with VMWare	System z
Servers	350	45	1
Memory GB	700	720	352
Cores	700	360	24
Software licenses	742	352	40
System administrators	35	18	5
Square feet of floor space	135	75	62
Kilowatt hours per year	3.2M	697K	127K

$7 million (USD) yearly savings

Annual savings provided by running a workload on System z versus distributed environments

Myth 3 : 99.999% Availability isn't that Important!

False!

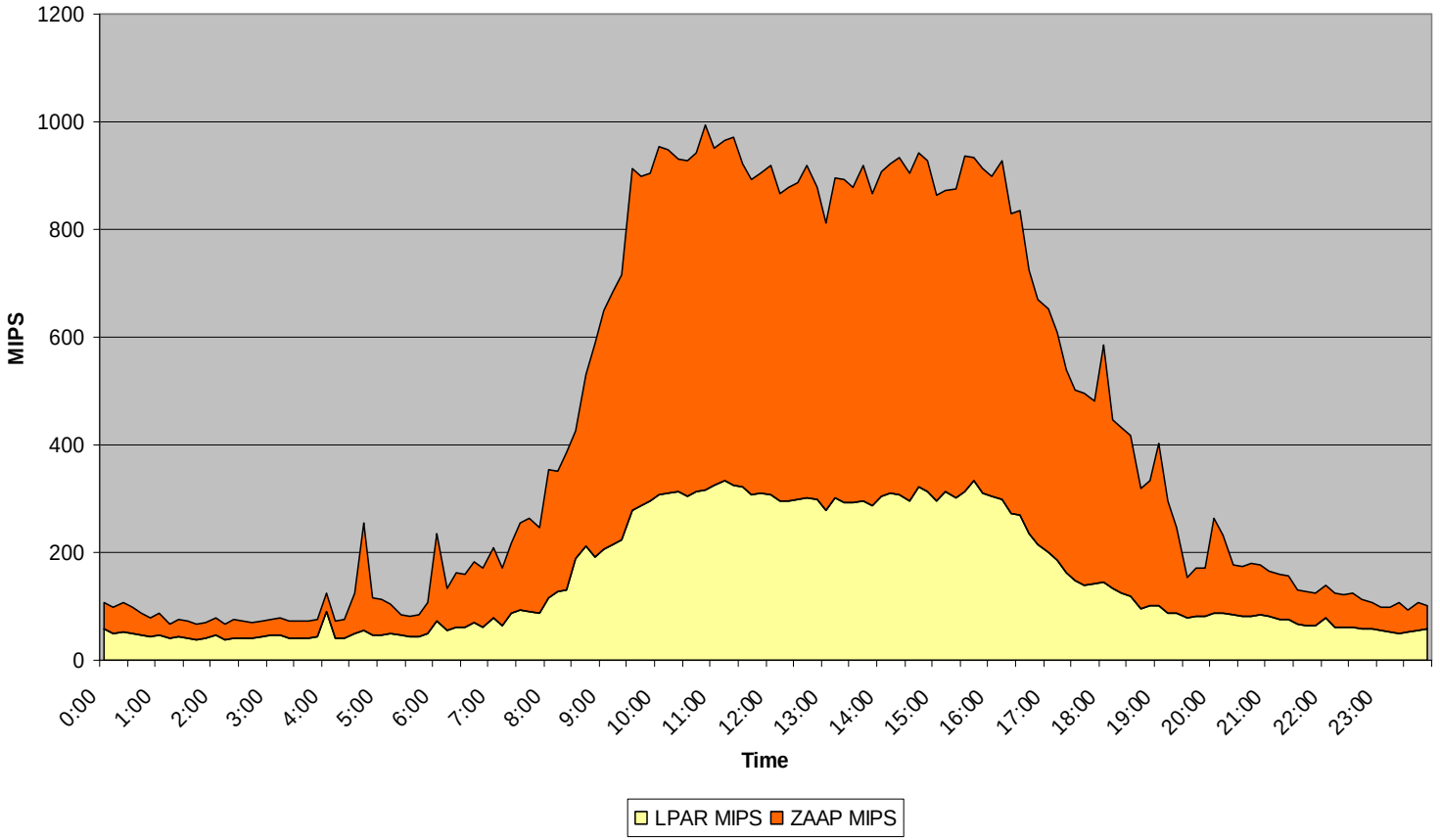
99.9% Availability Is Not Enough
The Decimals Matter!

Industry	Hourly Downtime Cost	Annual Cost
Brokerage Operations	$6,450,000	$56,502,000
Energy	$2,817,846	$24,684,331
Credit Card Sales Authorizations	$2,600,000	$22,776,000
Telecommunications	$2,066,245	$18,100,306
Manufacturing	$1,610,654	$14,109,329
Financial Institutions	$1,495,134	$13,087,374
Information Technology	$1,344,461	$11,777,478
Insurance	$1,202,444	$10,533,410
Retail	$1,107,274	$9,669,720
Pharmaceuticals	$1,082,252	$9,480,528

Sources: Meta Group; Fibre Channel Industry Association, Oct '00. Used with permission.

Myth 4 : Software on z/OS is too expensive

False!



"Costs reduced both by usage of zAAPs (66% offload achieved) and running Java on z10 (approx 8% reduction in CPU workload / NKS)"

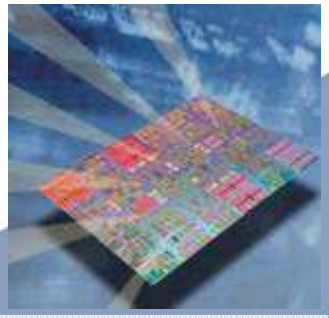
Source: Large AP Bank

Myth 5 : System z can't handle today's diverse Workloads!

False!

Internal Coupling Facility (ICF) 1997

Integrated Facility for Linux® (IFL) 2000



System z Application Assist Processor (zAAP) 2004

IBM System z Integrated Information Processor (zIIP) 2006

- **Eligible for zAAP:**
  - Java execution environment
  - z/OS XML System Services

Eligible for zIIP:

- DB2 remote access, XML, large parallel queries, utilities (index, sort, stats)
- ISVs
- IPSec encryption
- XML System Services
- Global Mirror (XRC)
- HiperSockets for large messages (e.g. DRDA)
- IBM GBS Scalable Architecture for Financial Reporting
- z/OS CIM Server
- zAAP on zIIP

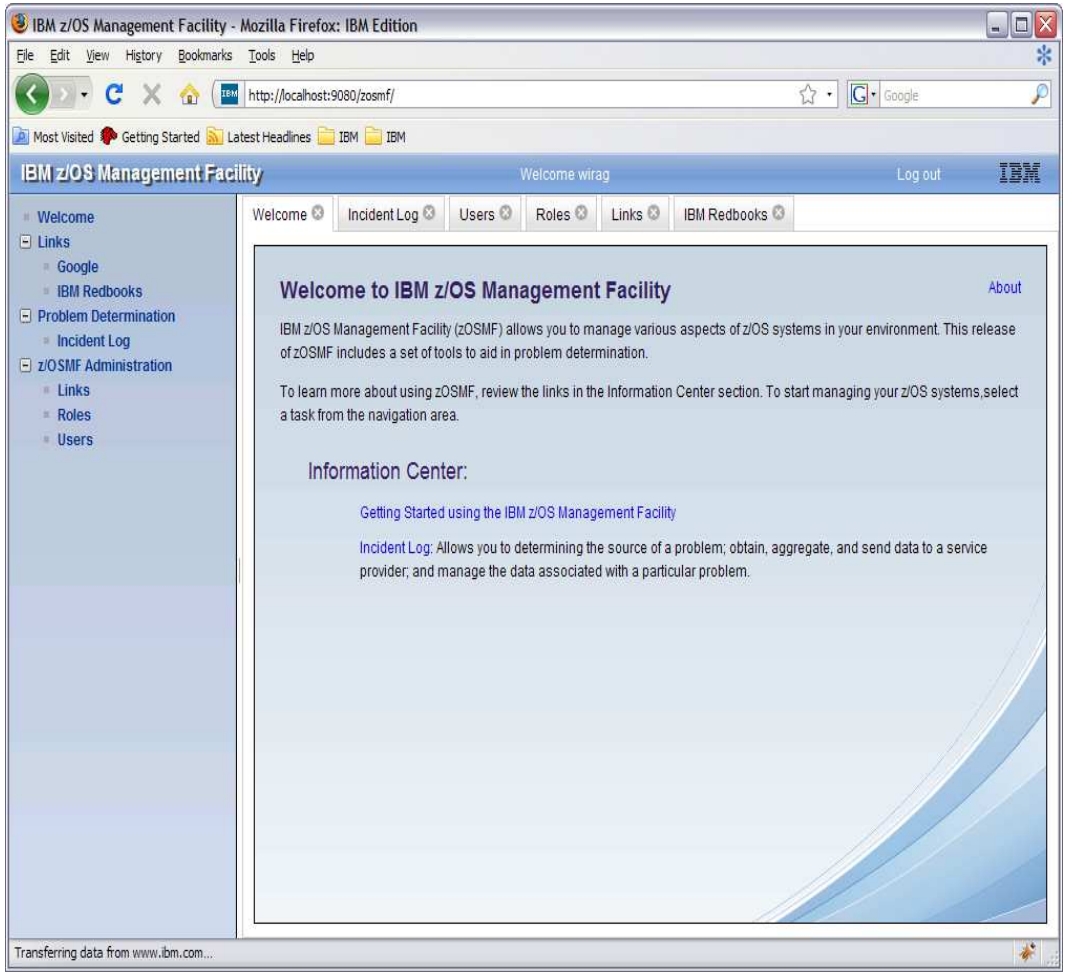
Specialty Engines

Myth 6 : System z is too hard to Manage

False!

- Web-based Management console for z/OS
- Zero priced feature of z/OS
- Helps new System Programmers and Administrators manage z/OS more easily
- Attracts new talent!

IBM z/OS Management Facility



Java and WebSphere on System z

Impact2012
Comes To You

Is the Java language on System z Different?

Is there anything special that needs to be considered when designing and writing Java code to run on System z or z/OS?

Answer:

Clear enough? ☺

Java is Java

The point of an open standard application platform is to eliminate platform dependencies

Two points:

1. **That wasn't always the case … earlier we had ASCII / EBCDIC issues. No more.**
2. **There are poor coding practices that can make bringing applications to z/OS problematic …**

WebSphere : A very Important Starting Concept

This point can't be stressed enough -- the differentiation is *not* in the open standard specification support offered. *That is common across platforms!*

"WebSphere is WebSphere" above the specification interface line

Java EE, Java SE, EJB, Servlet, JSP, JDBC, JCA, JMS, Web Services, etc.

Open Standard Specification Interfaces

How it's *implemented* is dependent on the platform … its features, functions, attributes and qualities of service

Starting with V6.0 the code base merged into one with a single source

Problems with code divergence solved. Code has ability to detect platform and invoke platform-specific exploitation as appropriate.

Key Benefits of Alignment Across Platforms

The alignment of specifications across all the IBM platforms brings several key benefits to you and your business:

Development and test

Mid-tier application hosting

Mainframe exploitation application hosting

- Avoids costly rewrite of applications to change platforms; reduces the testing effort
- Ability to promote applications “up the ladder” without concern about loss of interface function
- Ability to architect application designs that span multiple platforms without having to make sacrifices based on the platform
- Ability to settle on a common set of application tooling across all platforms
- Ability to have an essentially common management interface across all the platforms

Minor **differences exist in areas related to platform specifics such as starting servers. More in a bit.**

Impact2012
Comes To You

These benefits are intended … this is why “WebSphere is WebSphere” across the platforms

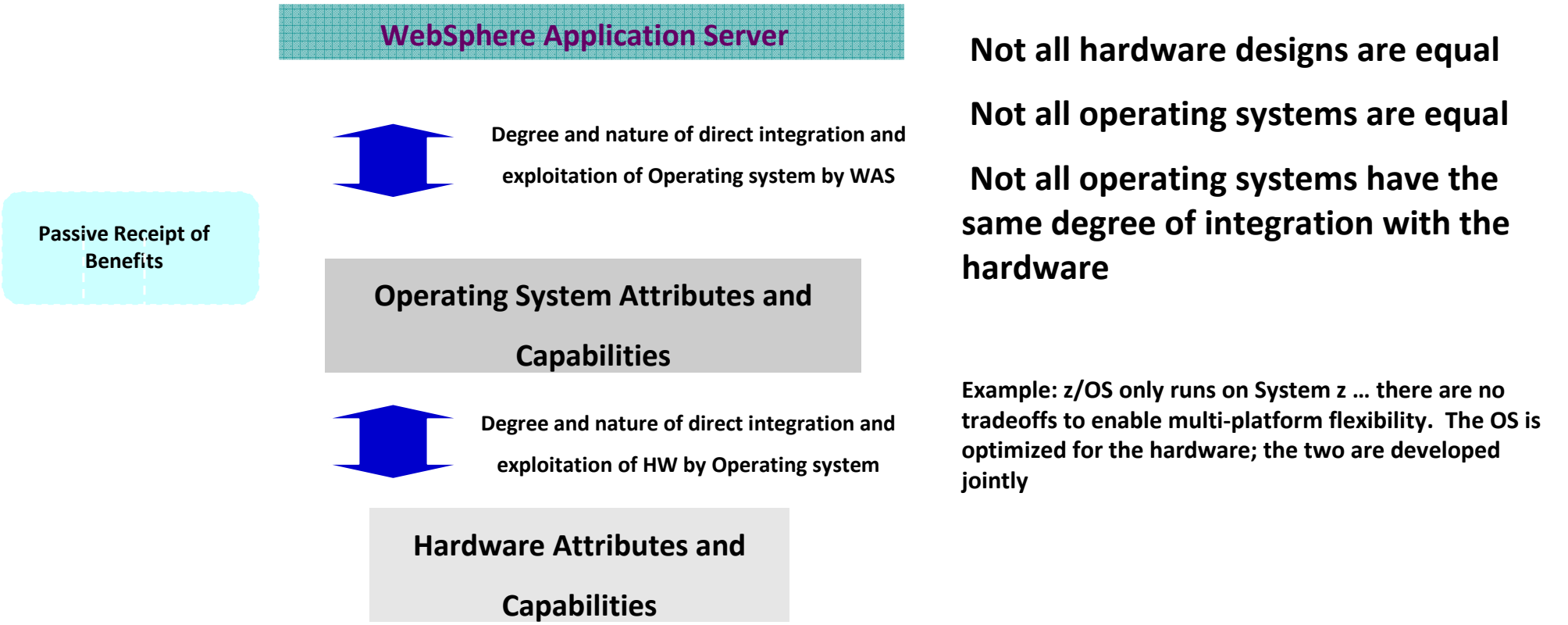
The Issue of Platform Exploitation

Establishing key concepts related to platform exploitation

Impact2012
Comes To You

Multiple Levels of Exploitation Taking Place

- We need to understand that there are benefits from the hardware design, benefits from the operating system design, *and benefits from the integration between the two*



Not all hardware designs are equal

Not all operating systems are equal

Not all operating systems have the same degree of integration with the hardware

Example: z/OS only runs on System z … there are no tradeoffs to enable multi-platform flexibility. The OS is optimized for the hardware; the two are developed jointly

That does not mean System z and z/OS are appropriate for all cases

Nor does it mean other platforms and operating systems can do what System z and z/OS can do

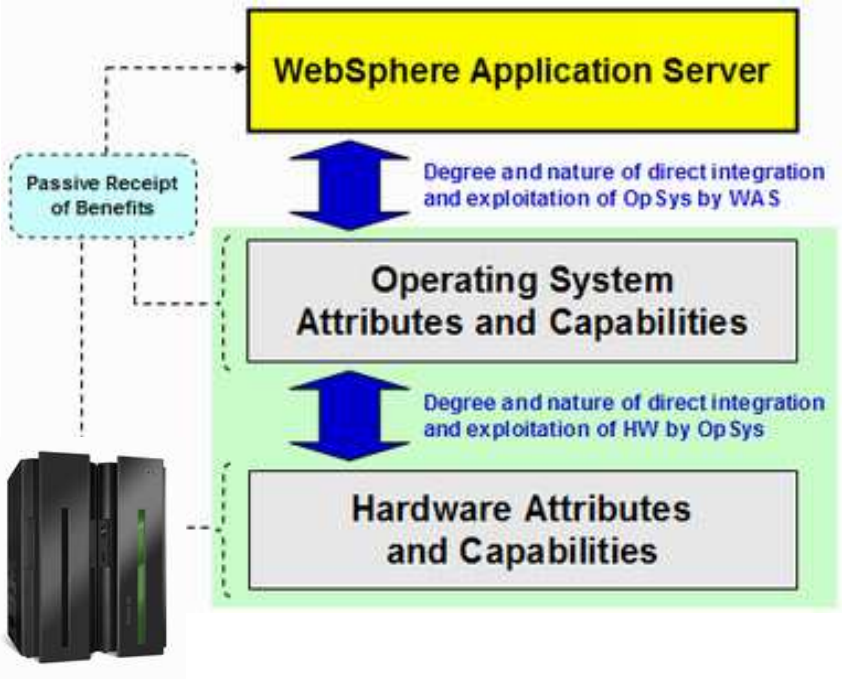
Passive Exploitation Benefits

Benefits derived by WebSphere Application Server by virtue of running on the platform

Impact2012
Comes To You

Passive Benefits Fall Into Several Categories

Programs that run on System z and z/OS receive passive benefits in a couple of different areas:



Hardware

- Inherent maturity and stability of design
- Redundancy and flexible updates
- Balanced design offers very high throughput
- Mature and proven virtualization through LPAR

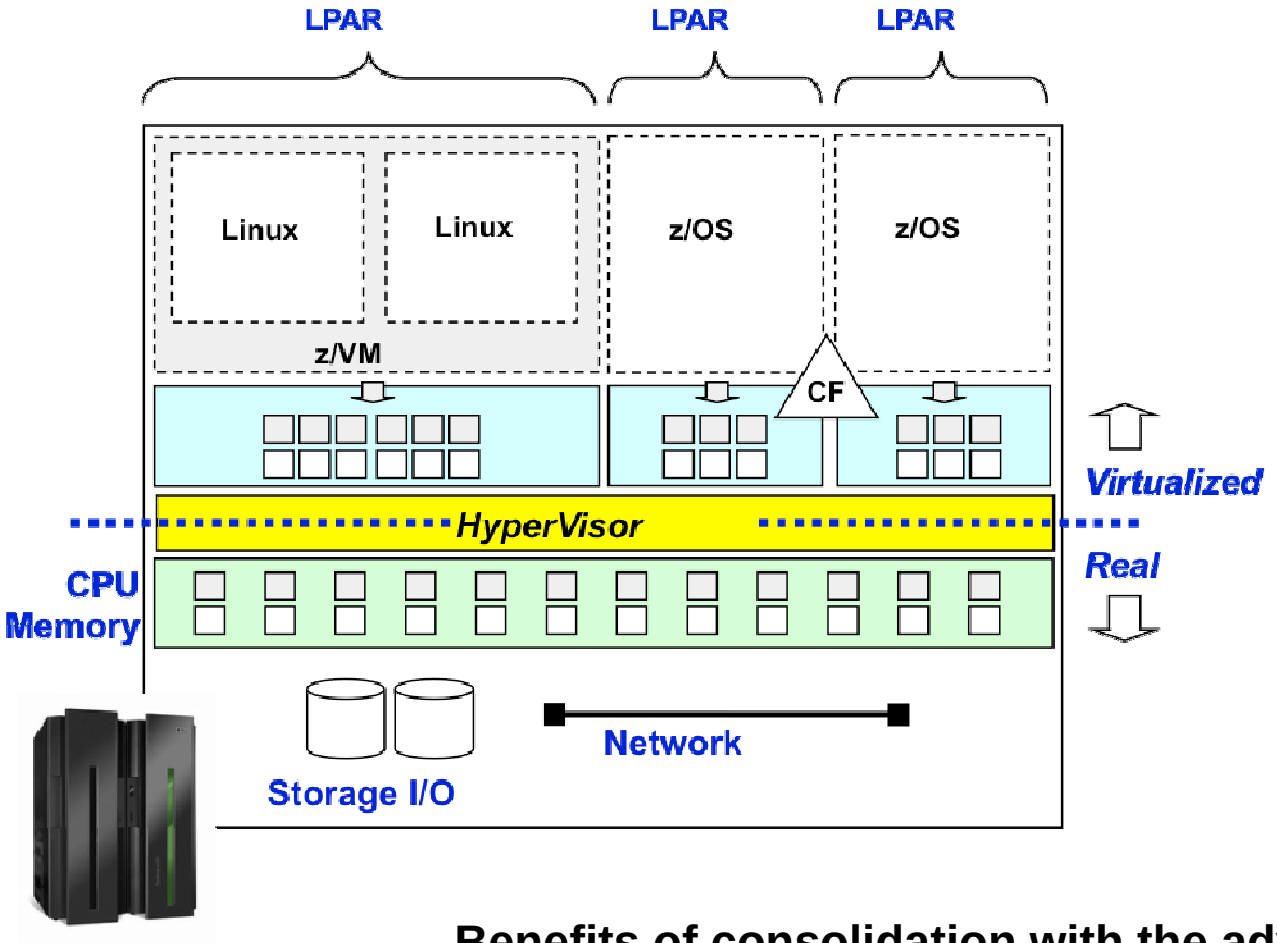
Operating System

- Tight integration with server hardware design
- Extremely mature architecture
- Storage protection
- Workload Manager (WLM)
- Intelligent Resource Director (IRD)
- Local TCP optimization
- Mature systems management tools
- Proven disaster recovery capabilities

Let's explore some of these at a high level

Hardware Virtualization -- Logical Partitions (LPAR)

An extremely mature virtualization technology that allows real resources to be shared across multiple logical partitions, each entirely separate



Proven virtualization technology

- Years of proven reliability
- Partitioning of HW into logical partitions
- Further virtualization using z/VM with guest machines

Each LPAR entirely separate from the other

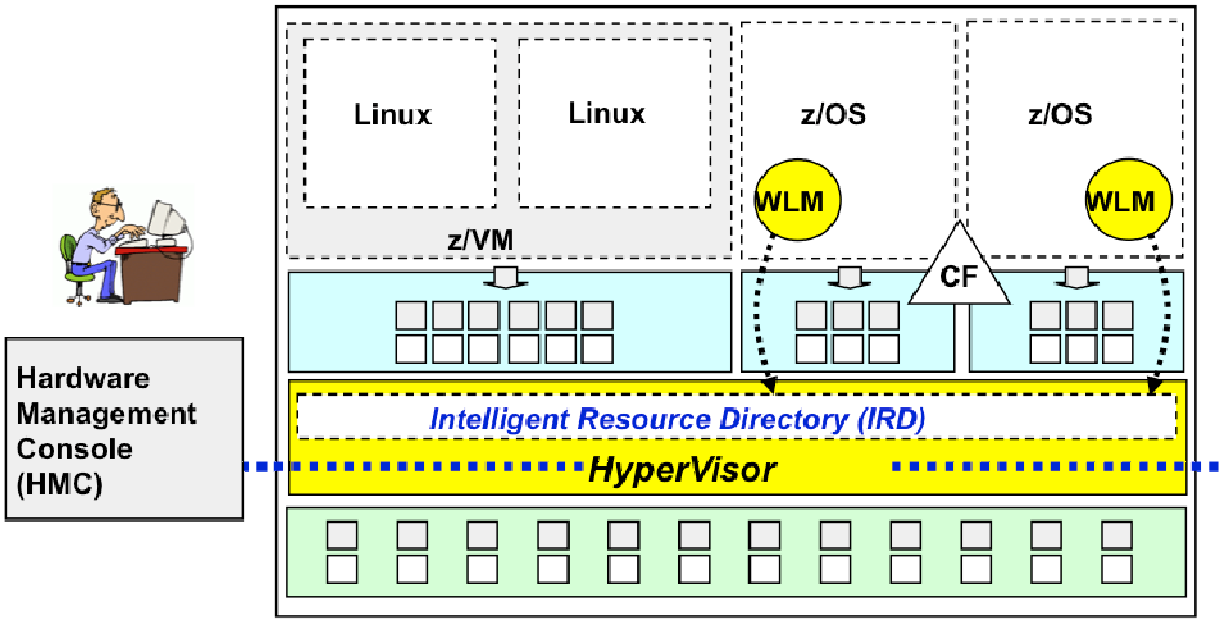
- Hypervisor protects one LPAR from monopolizing resources above what its allocated
- Complete memory isolation, so no overlay concerns
- Complete operating system isolation, so all elements of OS instances separated
- Complete network isolation, so no concerns about security breaches

Benefits of consolidation with the advantages of isolation

That's what virtualization is all about. The difference is one of maturity and capability. The technical differences between virtualization approaches can become a complex topic quickly. Point here is that System z LPAR has a proven production track record

Dynamic Modification of LPAR CPU Resource Allocations

Manual and automatic …



- Non-disruptively add CPU to the machine and assign to LPAR
- Allow IRD to dynamically move CPU between LPARs
- Dynamically vary I/O capacity across LPARs to solve bottlenecks
- With z/OS you may have WLM advise IRD to reallocate CPU and I/O between LPARs in the Sysplex

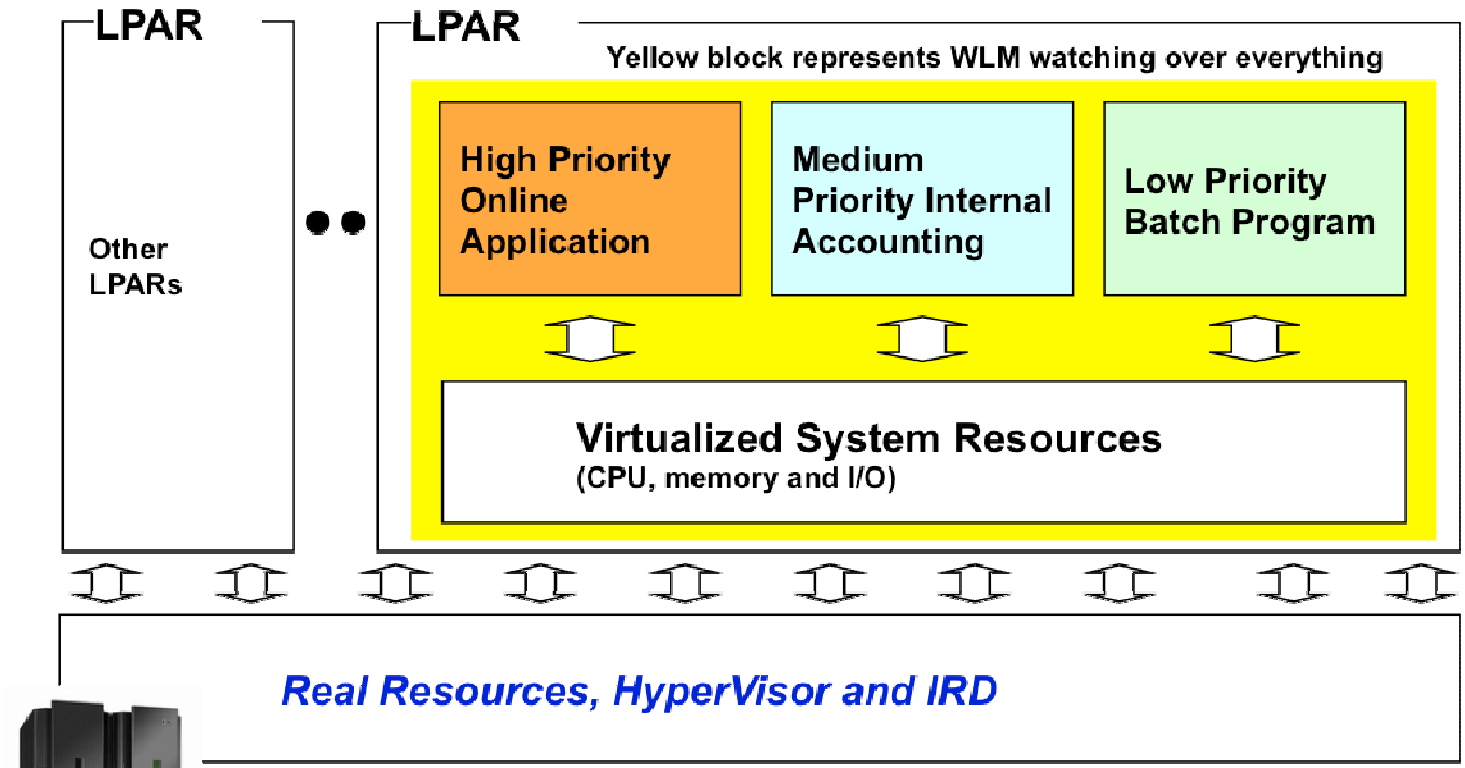
The message here is one of dynamic flexibility.

The pace of change is increasing … rigid designs hinder rapid exploitation of opportunities

System z LPAR technology coupled with z/OS WLM provides a proven flexible and dynamic environment

Workload Manager (WLM)

A component of the z/OS operating system, WLM keeps close watch on key system metrics and manages resources towards meeting your defined goals



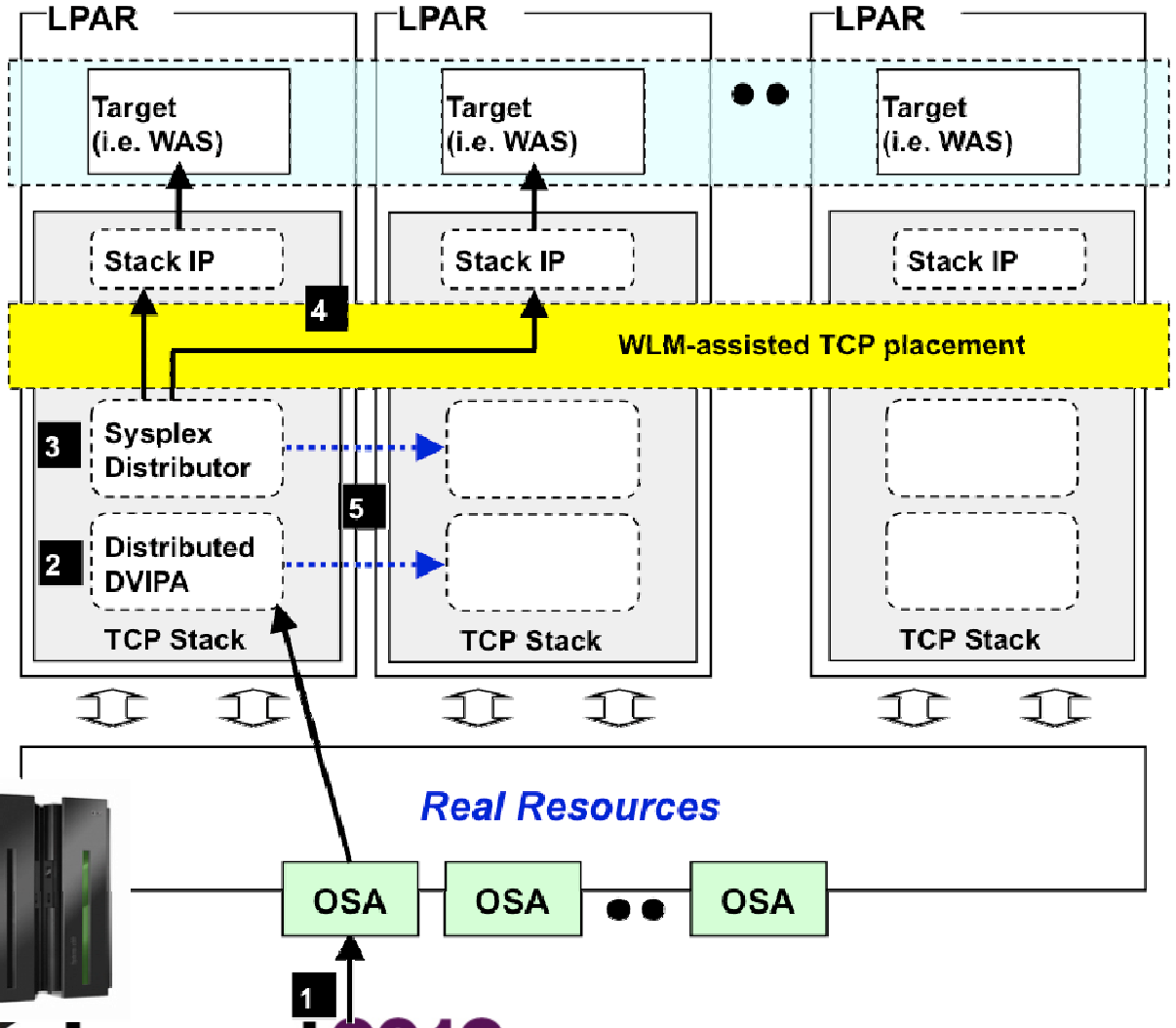
Five pieces to this:

1. WLM's real time monitoring of the overall system resource utilization
2. The WLM service level goals you've defined that determine how WLM will manage resources
3. WLM's comparing your service level goals against the actual system performance on a program by program basis
4. WLM's reallocating resources within the LPAR to make sure goals are met
5. WLM advising IRD if resource allocation across LPARs is needed

- It's a very sophisticated system monitoring and control mechanism
- It has matured over the course of *years* to be as reliable and effective as it is
- Other solutions claim to offer "workload management" but are often rather weak in function compared to how z/OS WLM operates

Virtual IP and Sysplex Distributor

Is a function of TCP on z/OS which allows you to "hide" duplicated resources behind a single IP address with WLM-assisted TCP connection placement



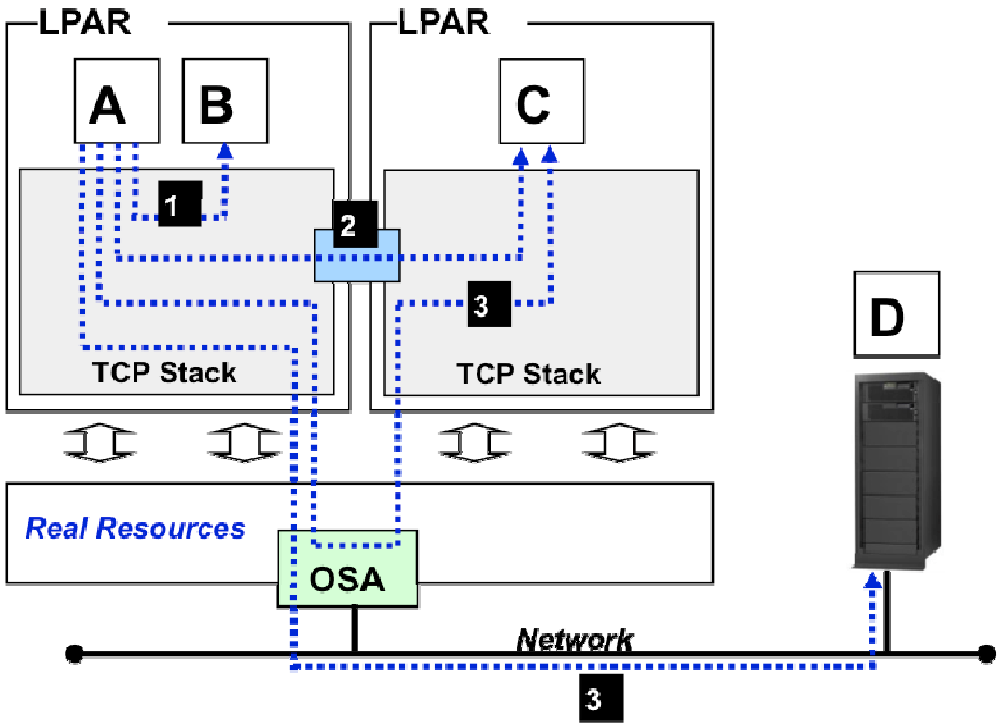
What's going on in this picture:

1. Clients in the world point themselves at a "generic" IP host name. Routers resolve that to one of the OSA adapters on the machine.
   - Note: there are ways to have redundant OSA adapters for availability
   - Note: it's not shown on this picture, but WLM can also advise some off-board Cisco routers.
2. Request is mapped to the TCP stack in the Sysplex that's hosting the Distributed Virtual IP (DVIPA) generic host.
3. Sysplex Distributor function determines which of the potential target LPARs is the best candidate to receive new work at that point in time.
4. TCP connection is made between client and the target
5. In the event of an outage of the hosting LPAR or TCP stack, the DVIPA and Sysplex Distributor functions automatically move to a defined "next in line" stack.

This is transparent to the application … it is passive in this process

Local TCP Optimization

z/OS is smart … it knows when client and target are on the same TCP and it optimizes the request with minimum code path employed



1. **Same LPAR**

   Request resolved within the TCP stack. Never gets to wire. Doesn't even get to the OSA adapter. (Also known as "Fast Local Sockets")

2. **Different LPAR, HiperSockets**

   Request flows memory-to-memory via HiperSocket network, which is a virtual network implemented by Hipervisor.

3. **Different LPAR, *not* HiperSockets**

   Request flows to OSA, but does not touch the wire. No short loop cables. Request stays in OSA microcode and then up to other LPAR.

4. **Off System z**

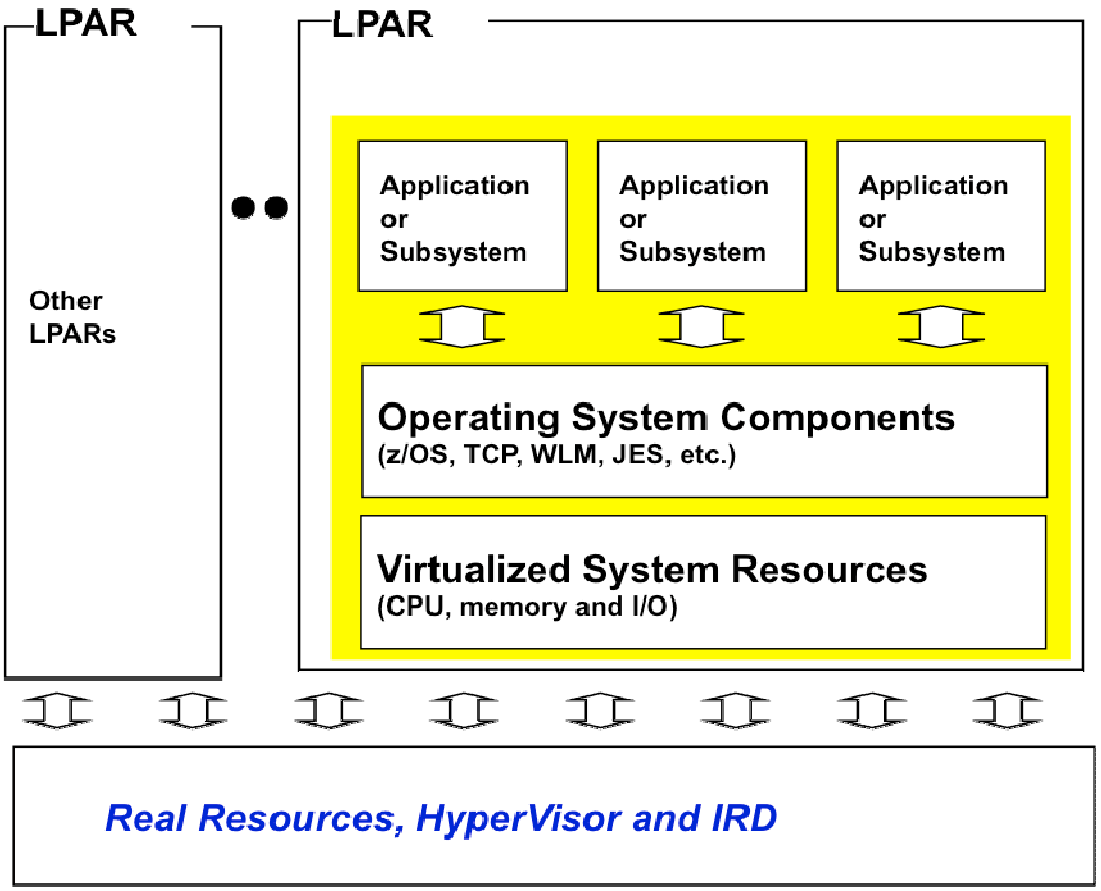
   Here System z has no choice but to go to the wire.

This can make a measureable difference

Network latency adds up as workloads scale

Resource and System Monitoring -- RMF and SMF

In order to manage your server environment effectively and efficiently, you'll need to understand who's using what and when



SMF

- A facility that components may use to write records to a system database.
- Those records may then be used to analyze system usage for:
  - Capacity planning
  - Performance planning
  - Accounting and chargeback

RMF

- Another facility that writes SMF to report on key system activities.
- Invaluable for planning and investigation of issues

If the platform is more manageable, then users of the platform derive indirect benefit from that

Note: WAS z/OS actively exploits SMF. We' ll see that coming up.

Active Exploitation Benefits

In two parts: IBM Java JDK for z/OS and WAS for z/OS

Impact2012
Comes To You

The JDK for z/OS

"Java is Java" but not all JDKs are the same, and not all JDKs are fully aware of and exploit the underlying platform

Examples of System z Exploitation by JIT

System z hardware has evolved over the years to have some very sophisticated underlying features. JIT in JDK for z/OS is written to exploit them:

The JIT is very much aware of the System z hardware features and exploits them directly for greater throughput and efficiency

z/OS-Specific Extensions to the JDK

In addition to the standard-compliant Java there are extensions to provide exploitations of System z and z/OS functions:

- ibm.com/servers/eserver/zseries/software/java/products/j6pcont64.html

The IBM 64-bit SDK for z/OS, V6 provides a full function SDK compliant with the SDK 6 APIs.

Documentation is available for content that is additional to the base.

- Security functions:
  - Java Cryptography Extension (IBMJCE)
  - Java Cryptography Extension in Java 2 Platform Standard Edition, Hardware Cryptography (IBMJCECCA)
  - Java Secure Sockets Extension (IBMJSSE)
  - Java Certification Path (CertPath)
  - Java Authentication and Authorization Service (JAAS)
  - SAF interfaces
  - Java Generic Security Services (JGSS)
  - Java PKCS#11 Implementation Provider (IBMPKCS11Impl)
- RMI-IIOP
- Java Record I/O (JRIO)
- JZOS - Java Batch Launcher and Toolkit

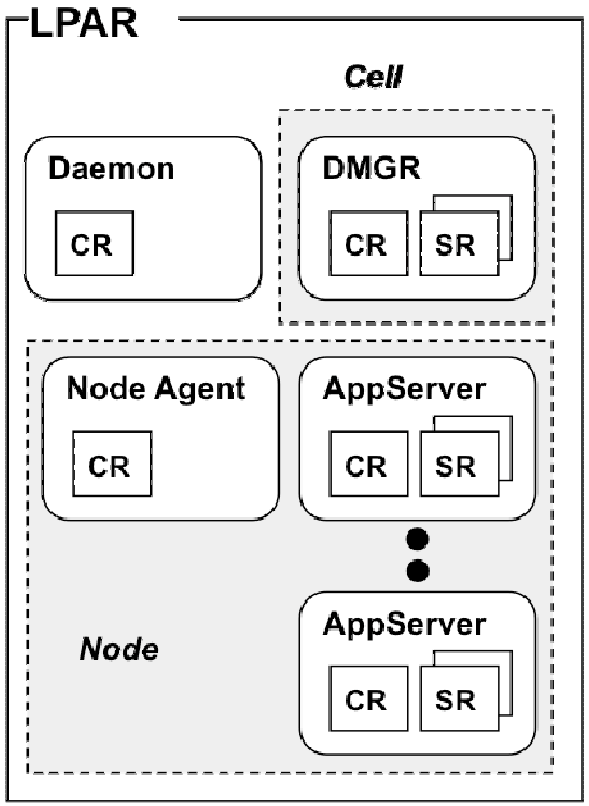
All content above is shipped with the z/OS SDK product and is zAAP eligible.

- Extensions to the JSE security standards to provide access to System z and z/OS facilities such as SAF and the Crypto Hardware
- Access to VSAM files, sequential files, PDS directories and the system catalog

Everything the standard JSE calls for, plus additional function that exploits the platform for added benefit to you

Exploitation of JES and Common z/OS Facilities

And that means that existing z/OS system programmers will be comfortable with the essential operations of WAS z/OS … it maps to their present skills

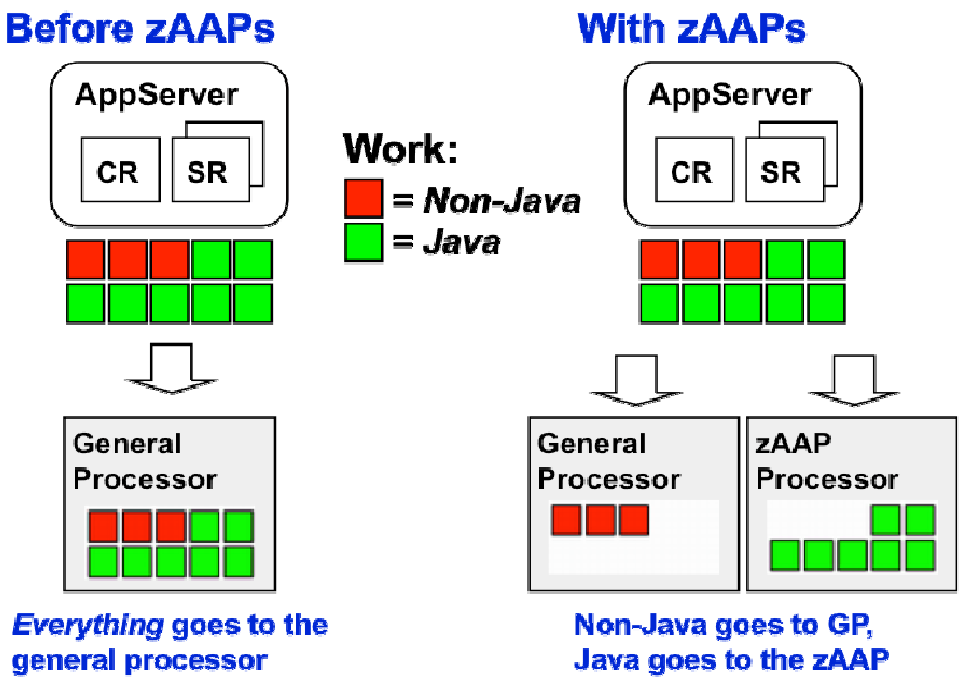


- WAS z/OS runtime implemented as a series of started tasks
- Standard JCL and START commands employed
- JCL START procedures maintained in PROCLIB
- Output written by default to JES
- JES manages output and storage
- Started tasks and address spaces displayable like any other
- Started and stopped like any other
- Able to use MODIFY commands for dynamic operations
- Configuration held in HFS or ZFS file systems
- Allows system automation tasks to control operations

This is all standard stuff. The key is that WAS z/OS was implemented to be compatible with existing z/OS skills, and to take advantage of existing z/OS facilities. WAS z/OS is *not* merely UNIX processes running in USS.

Exploitation of zAAP Specialty Engines

zAAP engines are Java offload engines. They enhance the financial picture of the z/OS platform, and they free up GP for other key subsystem processing



Keys to understanding value of zAAPs:

- zAAP processors have a considerably lower acquisition cost compared to GPs
- Offloading Java to zAAP frequently allows growing non-Java work to live within existing GPs, thus avoiding capital acquisition
- Monthly license charges based on capacity of the system can be influenced by the presence of zAAPs, which do not count towards charges

There are many technical details left unsaid here with respect to how they're configured, the rules for dispatching, when Java might go to GP, etc. Objective here was key points, not details.

This is really a function of the Java SDK and the dispatcher of z/OS.

The zAAP-enabled Java SDK is packaged with WAS z/OS, so WAS automatically takes advantage of zAAPs if they're present and configured

Exploitation of WLM

Intelligent Dynamic Capacity Expansion

The ability to increase the number of JVM instances based on WLM goals and configuration settings.

This is the “Controller / Servant” structure you may have heard about

Intelligent Workload Flow Control

An element of the Controller/Servant structure. Inbound work is queued and held, waiting for a thread to select it, based on importance and arrival. It’s a pull model rather than a push. Applications in JVMs take only what they can handle.

Intelligent Management of Mixed Work in Server

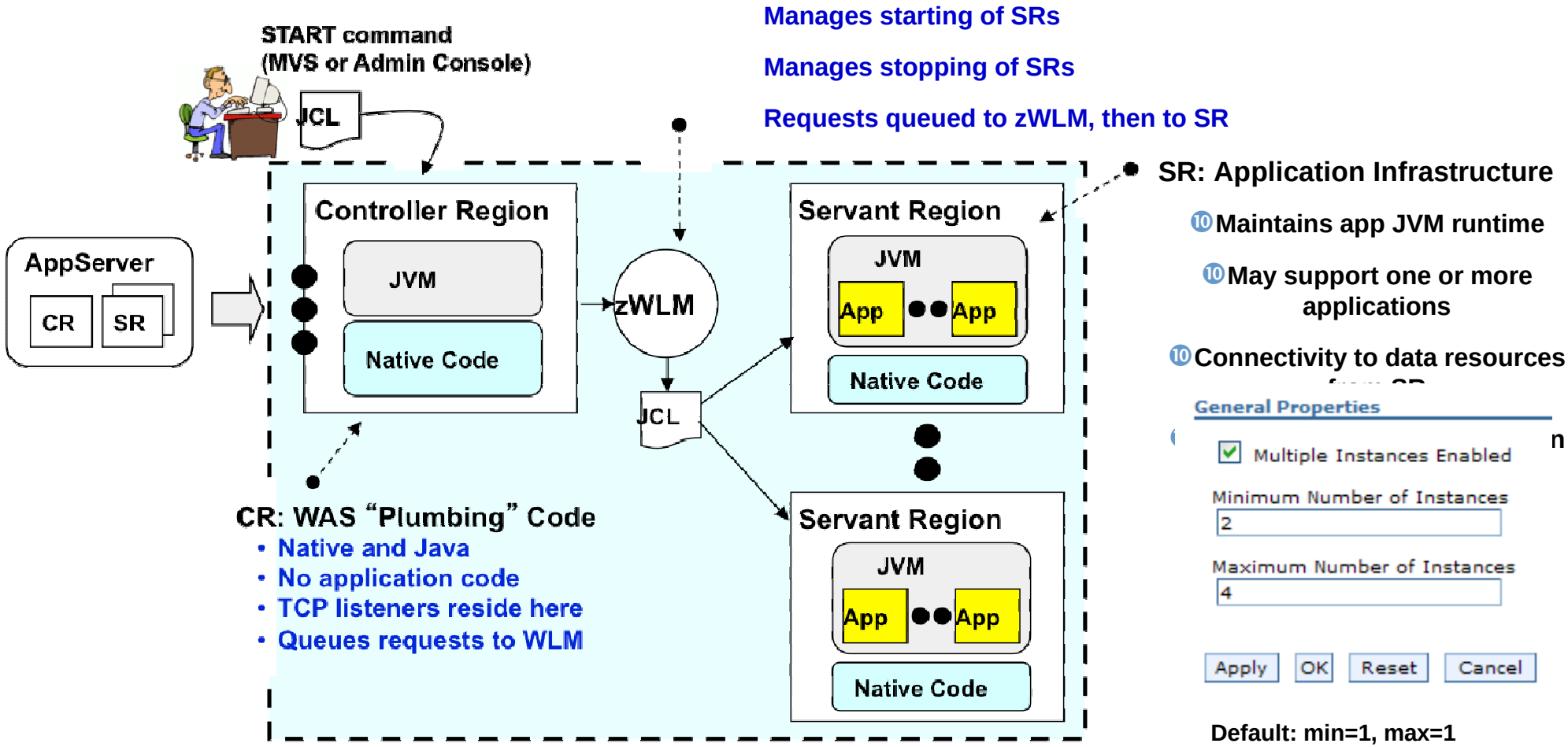
Multiple servants allows differently classified work to be placed in different servant regions. This allows WAS/WLM to understand what kind of work is in each and to manage system resources accordingly.

Intelligent Workload Routing Advice

WAS z/OS using WLM to determine where best to route certain kinds of work

The Controller / Servant Architecture

This is a unique architectural element to the WAS z/OS design. No other platform has this design because no other platform has WLM:**

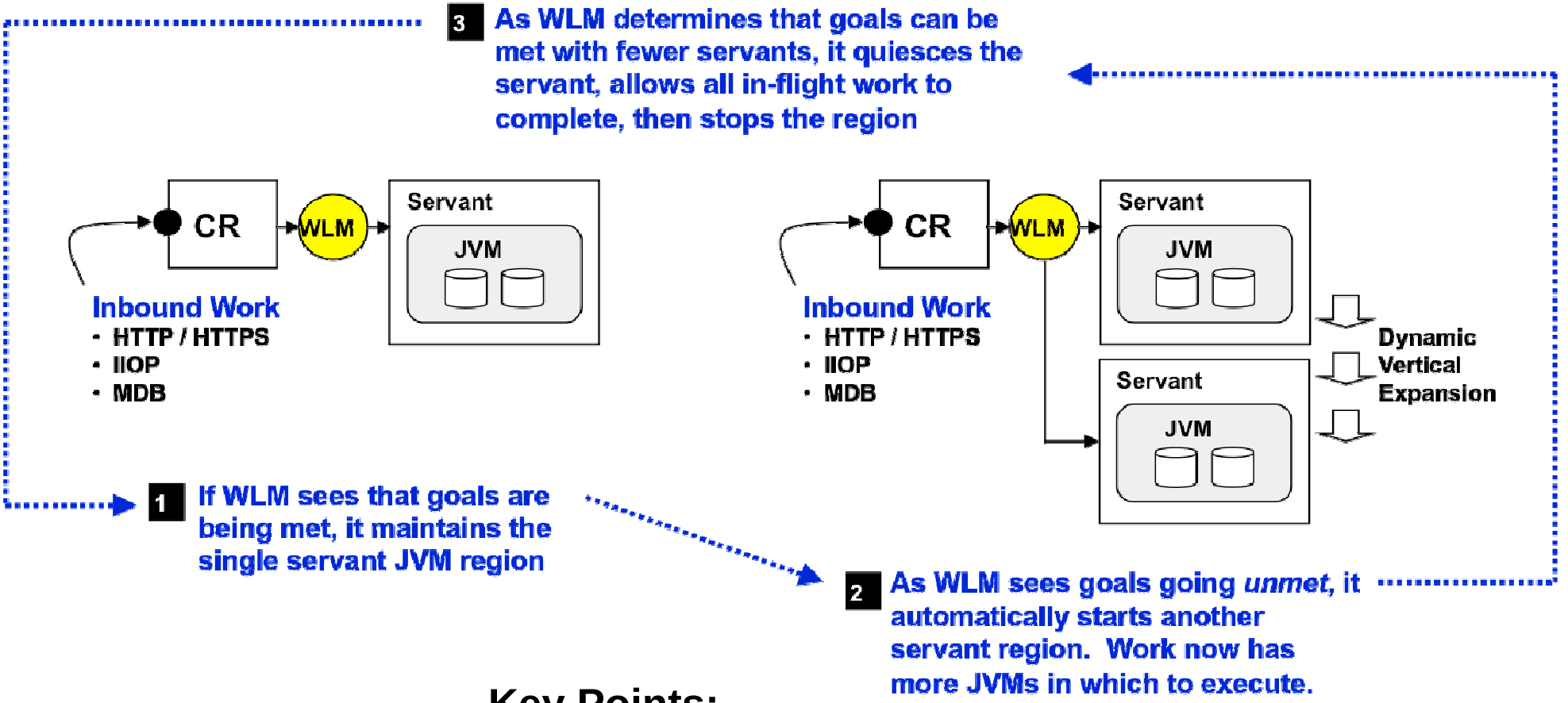


Let's now explore how this is accomplished …

** WebSphere on distributed uses the phrase "Workload Management" but it's not the same as zWLM

Intelligent Dynamic Capacity Expansion

This is the "vertical scaling" capability of the multi-Servant structure. If allowed, WLM will start additional servant regions if it sees unmet goals:

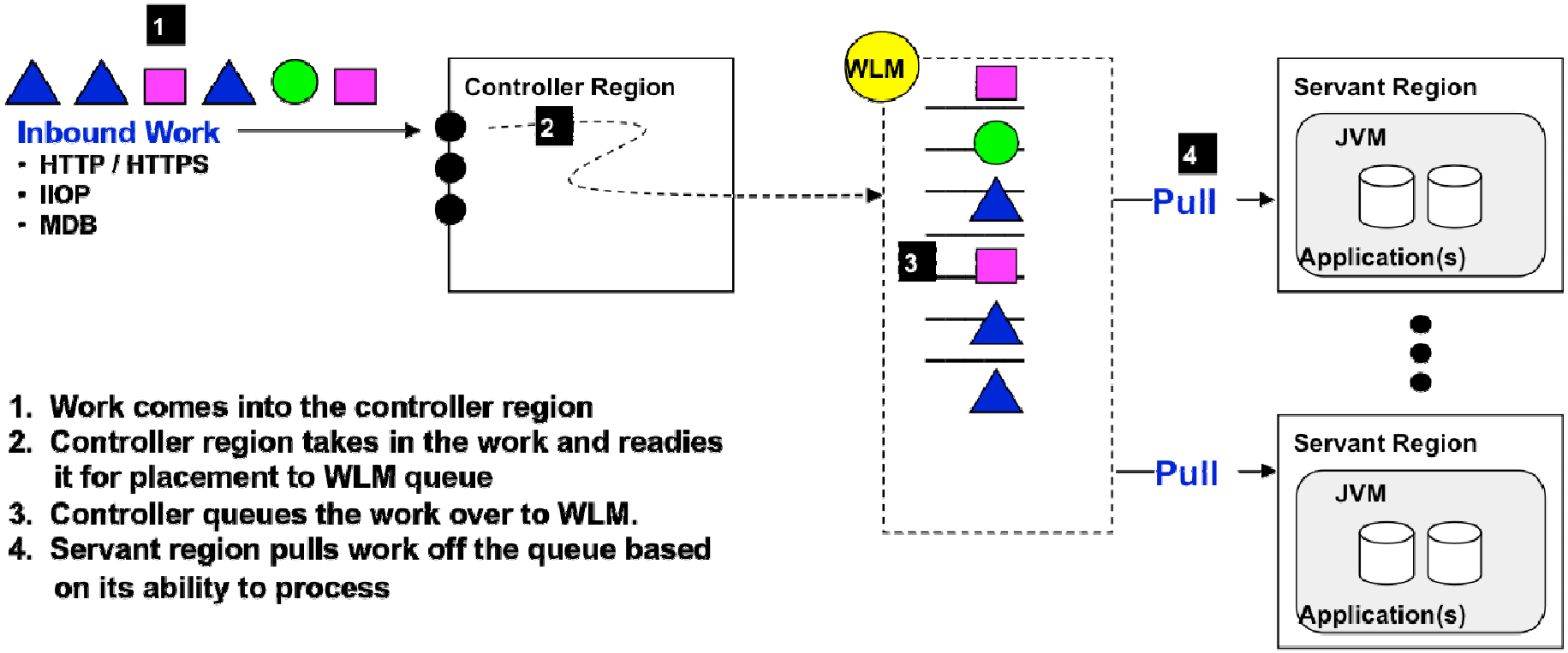


Key Points:

- The minimum and maximum number of servants is configurable. Default: Min=1, Max=1
- We see distributed WAS users trying to do something similar by configuring a "vertical cluster" to provide duplicate JVMs on a server box. Not quite the same -- no WLM assist of that

Intelligent Workload Flow Control

This is the WLM queueing mechanism that exists between the CR and the SR. It creates a "pull" model that prevents overwhelming an application JVM:



1. Work comes into the controller region
2. Controller region takes in the work and readies it for placement to WLM queue
3. Controller queues the work over to WLM.
4. Servant region pulls work off the queue based on its ability to process

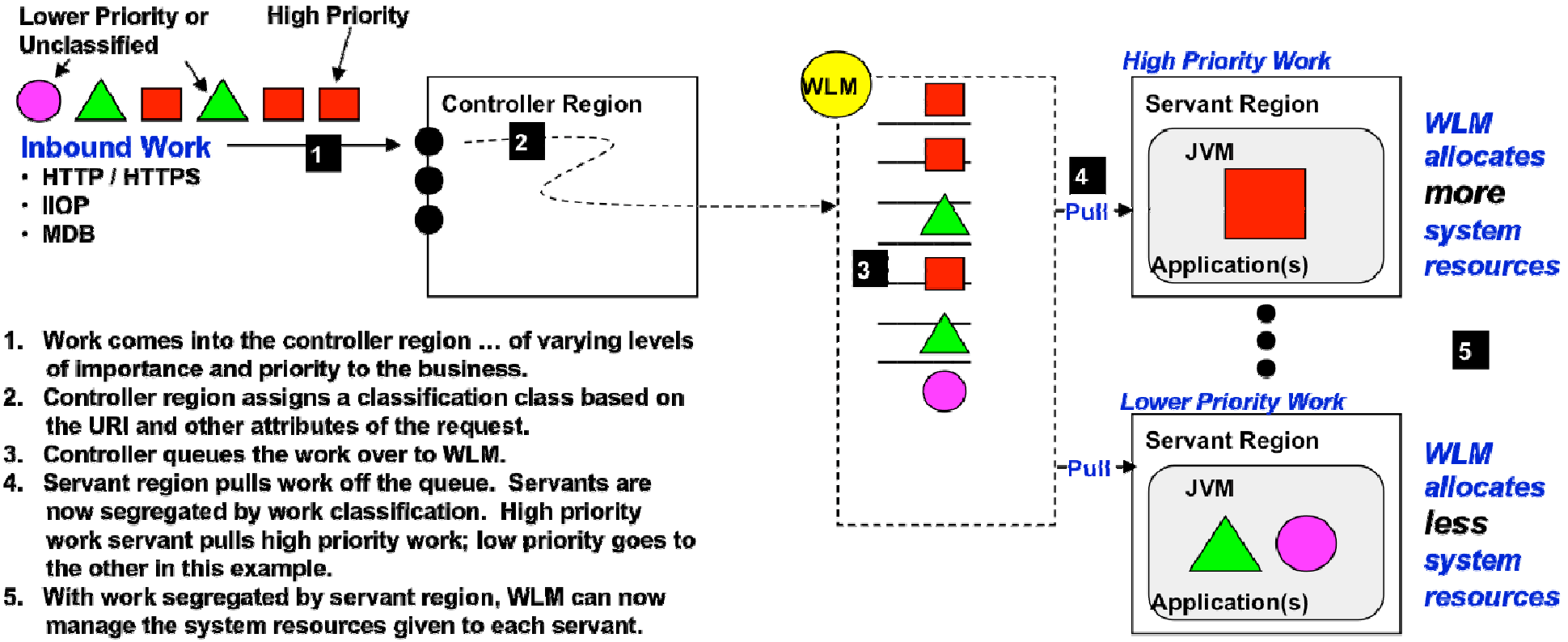
Servant can't be overwhelmed

Servant only takes what it can. Controller will take in and queue up what can't be handled immediately.

Intelligent Management of Mixed Work in Server

This involves inbound work being given a "Transaction Classification."

With that, the CR can direct work to servants and WLM can manage:



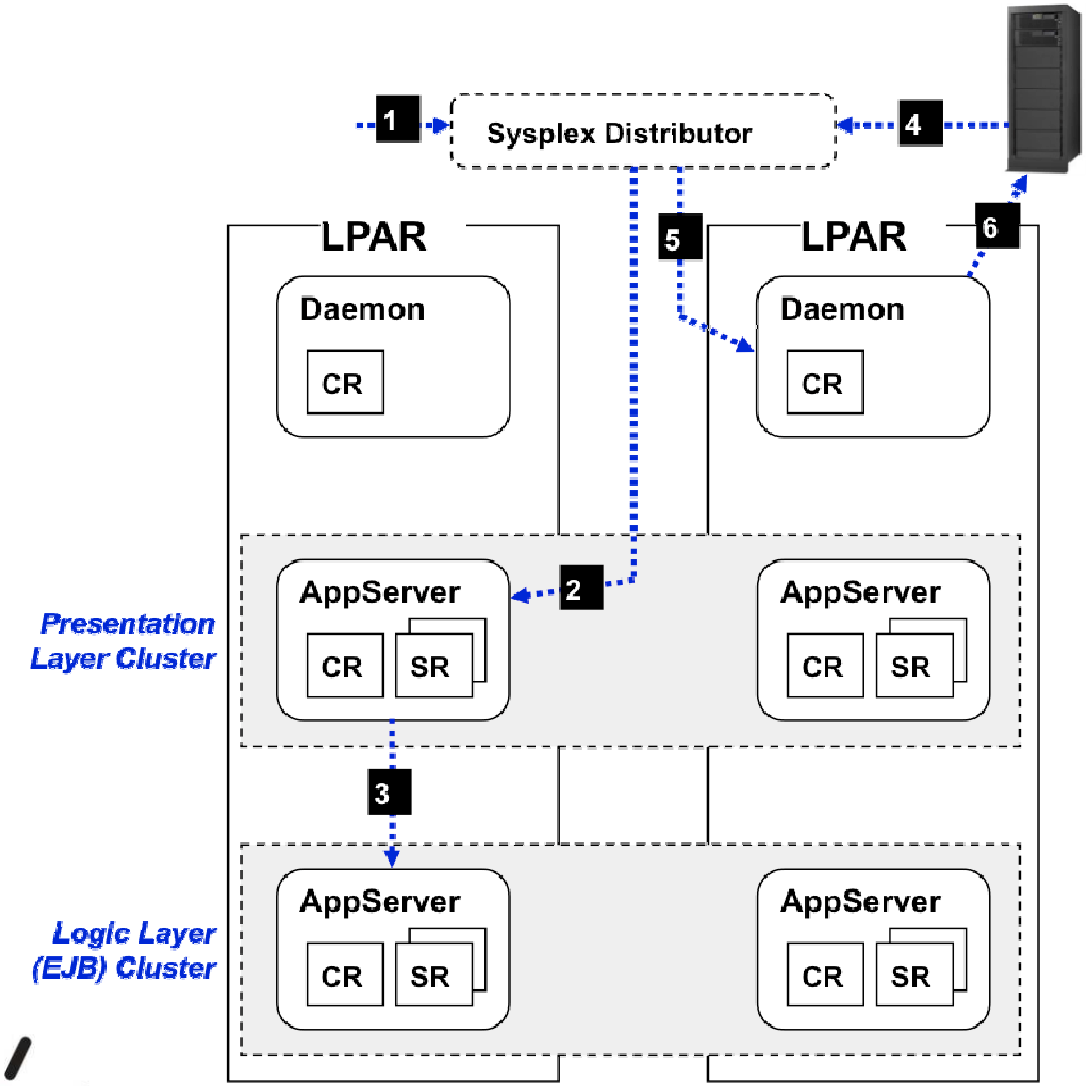
1. Work comes into the controller region ... of varying levels of importance and priority to the business.
2. Controller region assigns a classification class based on the URI and other attributes of the request.
3. Controller queues the work over to WLM.
4. Servant region pulls work off the queue. Servants are now segregated by work classification. High priority work servant pulls high priority work; low priority goes to the other in this example.
5. With work segregated by servant region, WLM can now manage the system resources given to each servant. High priority work gets more, lower priority less, all according to defined WLM goals.

Sophisticated Work Prioritization

On other platforms this can only be done by allocating work to separate servers. No WLM there to manage at this level.

Intelligent Workload Routing Advice

WAS z/OS relies on WLM to make routing decisions. We see this exercised in a couple of key areas:



Inbound HTTP or IIOP Work

- Work comes in over network aimed at DVIPA and Sysplex Distributor
- WLM advises Sysplex Distributor of presentation cluster member to place TCP request to
- For servlet-to-EJB flow (IIOP) WLM advises server, which then places the IIOP request to one of the members in the EJB cluster

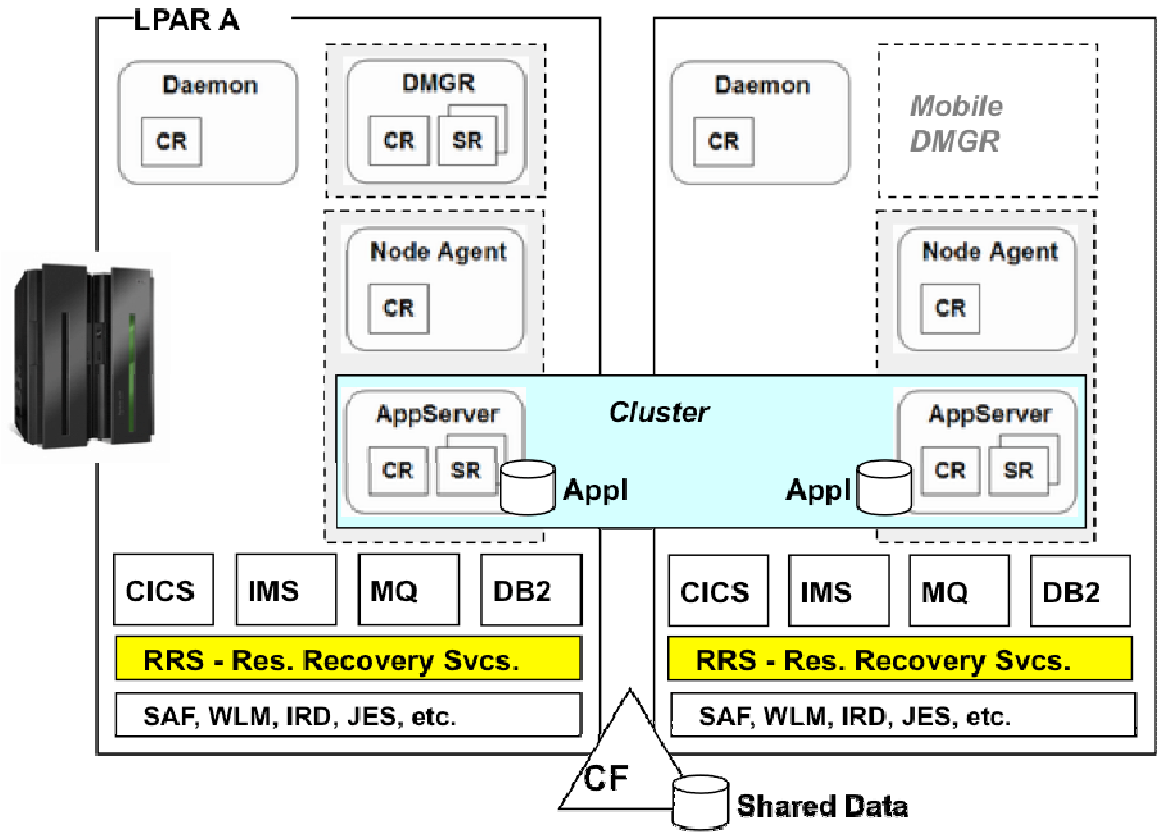
IOR Resolution

- External client seeks to use EJB deployed in WAS. It addresses itself to DVIPA/Sysplex Distributor for bootstrap (not shown on picture), then gets DVIPA host for Daemon location service.
- External client then come back to DVIPA/SD for Daemon location service. WLM advises Sysplex Distributor for best Daemon to place request.
- Daemon consults WLM for object location best able to service external client at that time. External client provided with host:port for EJB in the cluster.

The point here is that WLM plays a key role in the routing decisions made by clients and WAS itself for real-time routing

Exploitation of Resource Recovery Services (RRS)

Two-phase commit processing involves coordination of participants to make sure all are ready to commit. RRS plays that role in Parallel Sysplex:



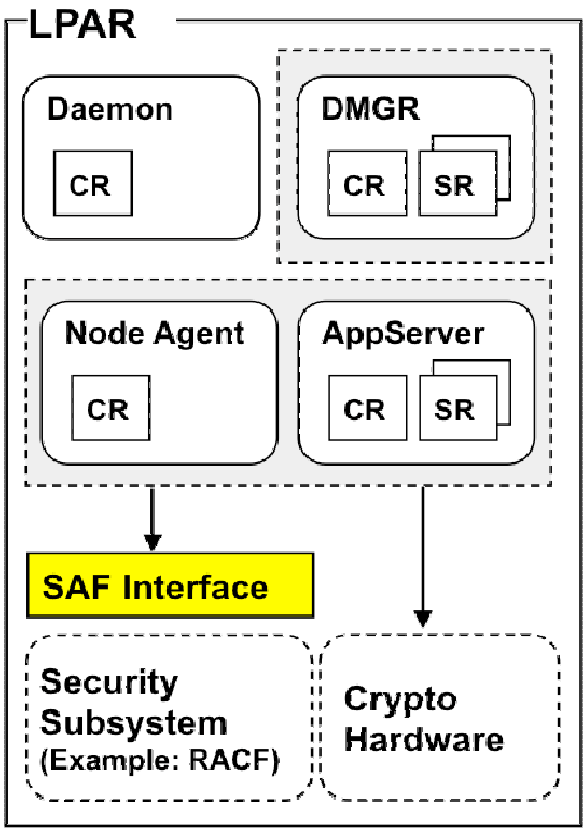
- We'll see this picture later when we discuss high availability
- WebSphere Application Server is a transaction manager … it is able to initiate a transaction and have other resource managers (DB2, CICS, IMS) participate in the unit of work
- For two phase commit processing, *someone* has to play the role of syncpoint coordinator
- On z/OS and Parallel Sysplex that someone is RRS, which uses Coupling Facility data structures and patented recovery algorithms to provide very efficient failed transaction recovery
- WAS z/OS registers with RRS, as do resource managers. RRS handles the two-phase commit coordination

Impact2012
Comes To You

Another case of "below the specification line" exploitation of existing z/OS and Parallel Sysplex technology to perform a task in an optimized manner for the platform

Exploitation of SAF and Crypto

SAF is a security interface; Crypto is a hardware-assist processor for encryption and key storage on the System z and z/OS platform



SAF Security Subsystem

- Sysplex-wide integrated security repository
- Single location for security artifacts rather than scattered model
- IDs, groups, keyrings, certificates, EJB role enforcement
- Local access … unlike LDAP, do not need to traverse network
- Extremely robust security model

Crypto Hardware

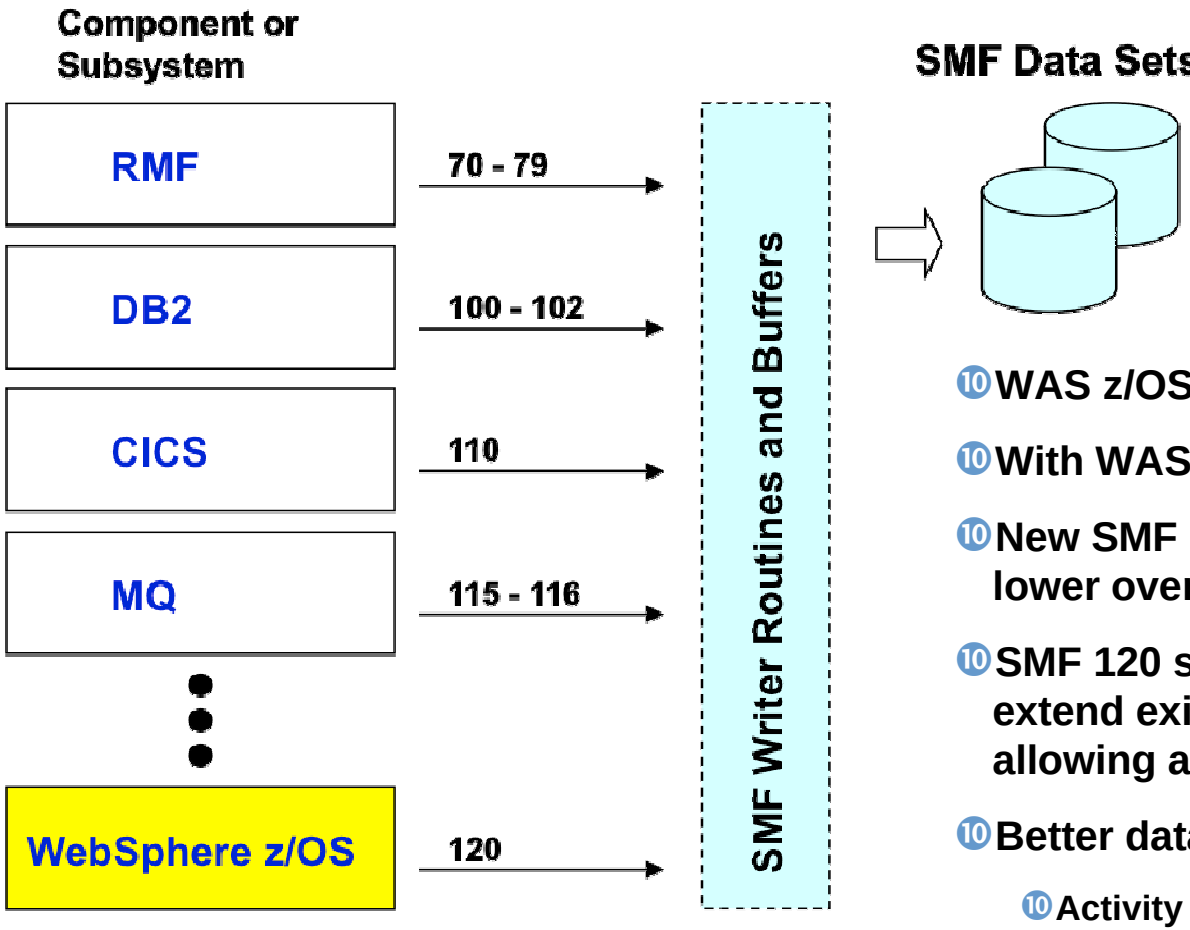
- Hardware-assisted cryptographic encryption and de-encryption
- Extremely secure private key store management

Properly configured, z/OS provides an extremely secure environment … many say the most secure available

Impact2012
Comes To You

Exploitation of SMF

- SMF is an activity recording facility of z/OS that allows subsystems to record key activity for analysis, management and accounting chargeback



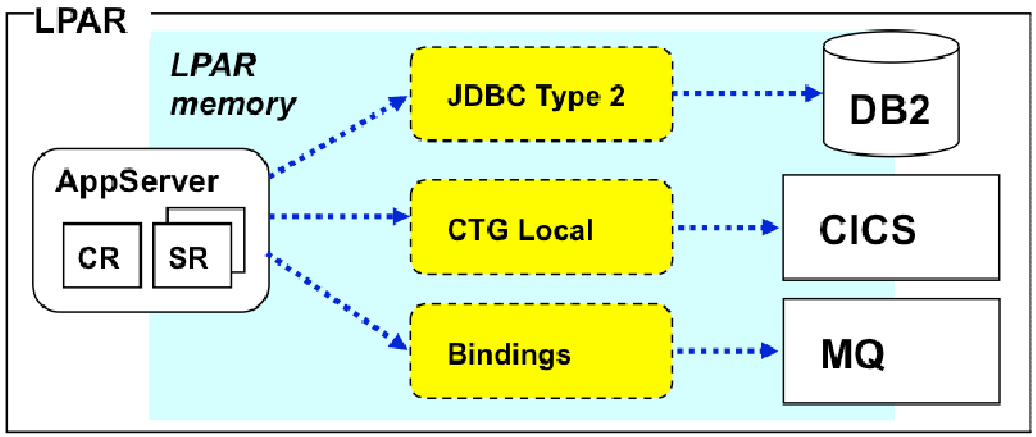
- WAS z/OS writes SMF 120 records
- With WAS z/OS V7, a new subtype was created: 9
- New SMF 120 subtype 9 provides better data with lower overhead cost
- SMF 120 subtype 9 and 10 records complement and extend existing SMF from other subsystems, allowing a far better picture of what's going on
- Better data available for …
  - Activity analysis
  - Usage statistics
  - Accounting chargeback

- http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101342

Exploitation of Cross-Memory Communications

Any time client and target are in the same LPAR, there's an opportunity for cross-memory exploitation. Let's look at a few examples:

Data Access

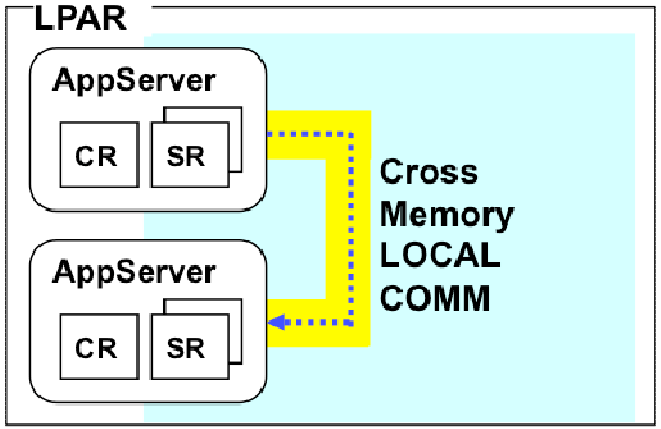


Benefits:

- Cross memory speed
- Security ID propagation (no alias)
- Exploitation of RRS
- Avoid serialization of parameters
- Avoids SSL overhead
- Single thread of execution

LOCAL COMM

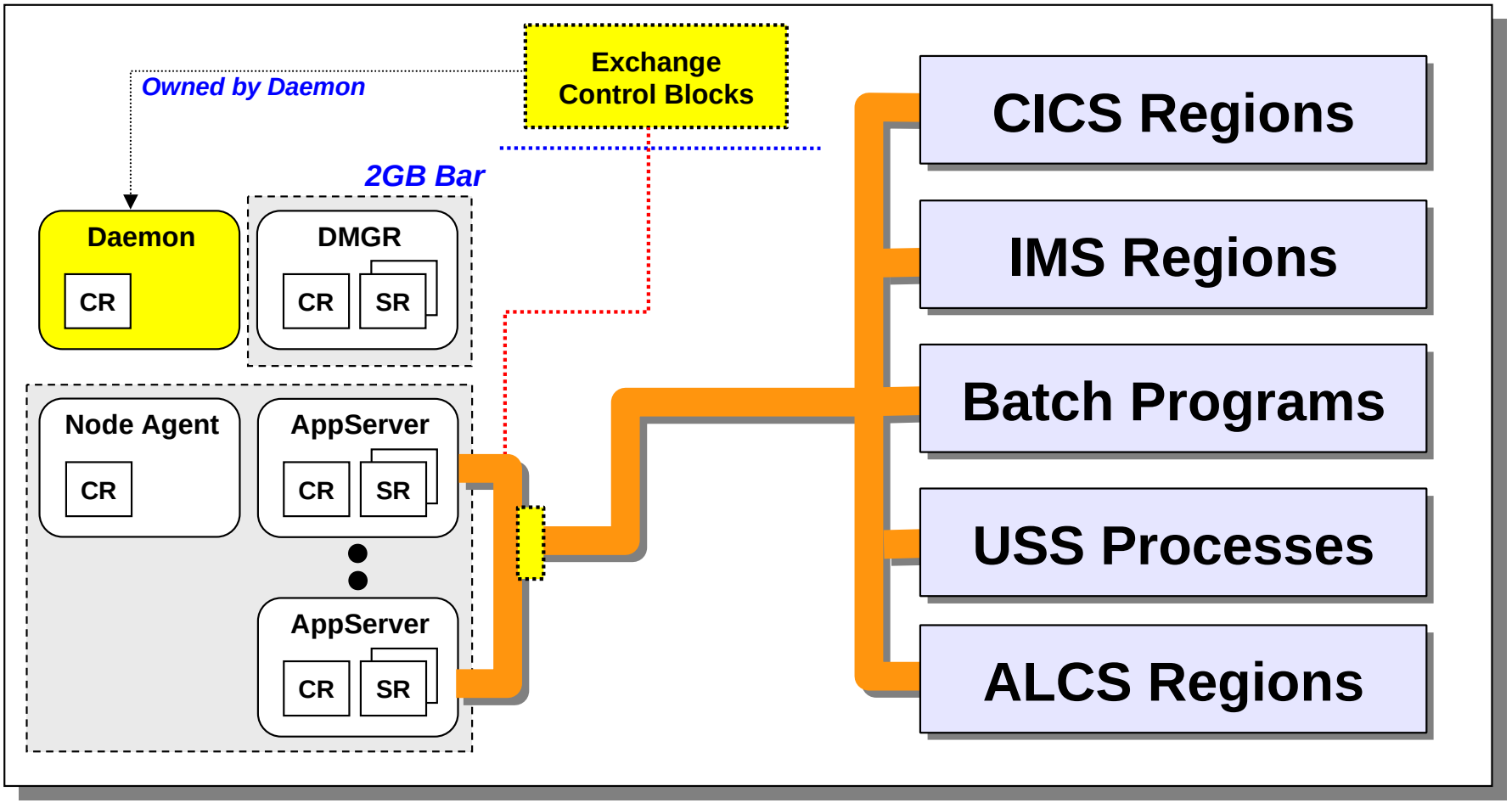
Used for IIOP flows between servers on the same LPAR.



Benefits:

- Avoids IP stack entirely
- Avoids SSL overhead
- Very fast, very secure

Cross-Memory+ : Optimized Local Adapters (WOLA)



Benefits:

- Based on Local Comm (z/OS exclusive)
- Bi-directional … WAS outbound or inbound to WAS (WOLA exclusive)
- CICS Security, Transaction, Workload context propagation (some restrictions apply)
- Faster than other local solutions

See WP101490 on `ibm.com/support/techdocs` for more

WAS and zEnterprise

Understanding the key differences and position the zEnterprise offerings

Impact2012
Comes To You

IBM zEnterprise System – Best in Class Systems and Software Technologies *ery*

Unified management for a smarter system:

zEnterprise Unified Resource Manager

- Unifies management of resources, extending IBM System z® qualities of service end-to-end across workloads
- Provides platform, hardware and workload management

The world's fastest and most scalable system:

IBM zEnterprise™ 196 (z196)

- Ideal for large scale data and transaction serving and mission critical applications
- Most efficient platform for Large-scale Linux® consolidation
- Leveraging a large portfolio of z/OS® and Linux on System z applications
- Capable of massive scale up, over 50 Billion Instructions per Second (BIPS)

Scale out to a trillion instructions per second:

IBM zEnterprise BladeCenter® Extension (zBX)

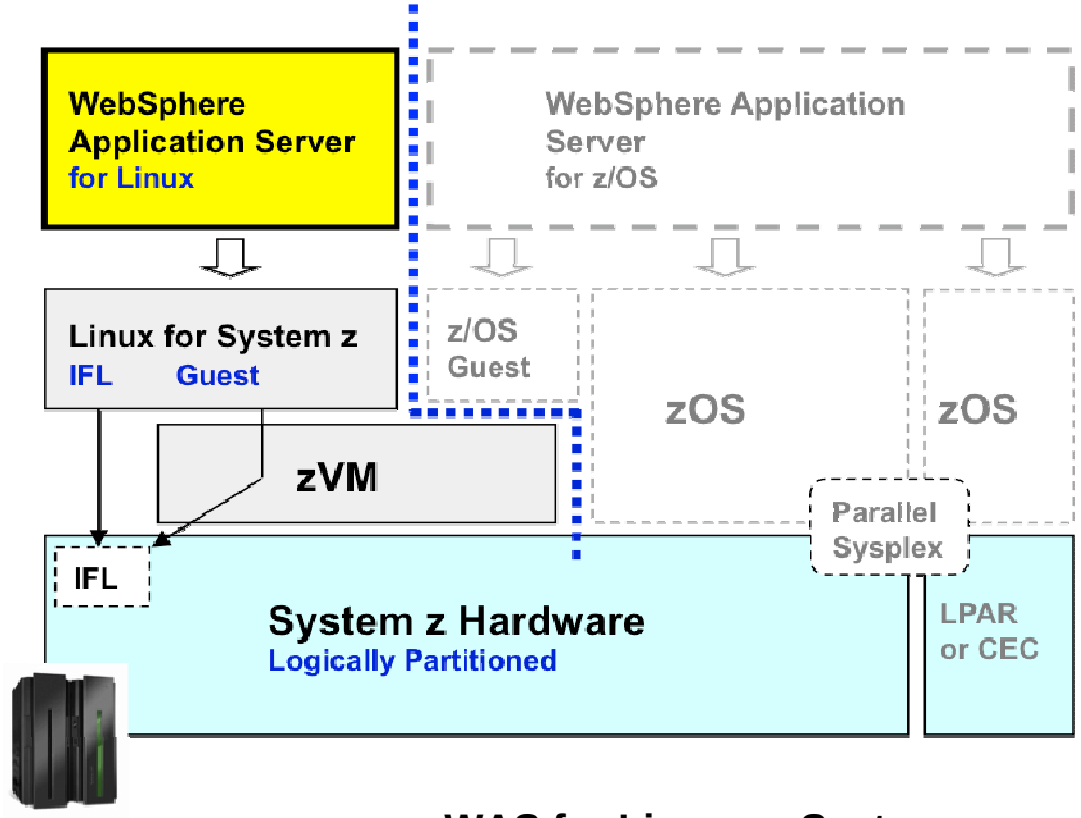
Selected IBM POWER7® blades and IBM System x® Blades[1] for tens of thousands of AIX® and Linux applications

High performance optimizers and appliances to accelerate time to insight and reduce cost

Dedicated high performance private network

Our Earlier Picture

We saw this picture earlier …



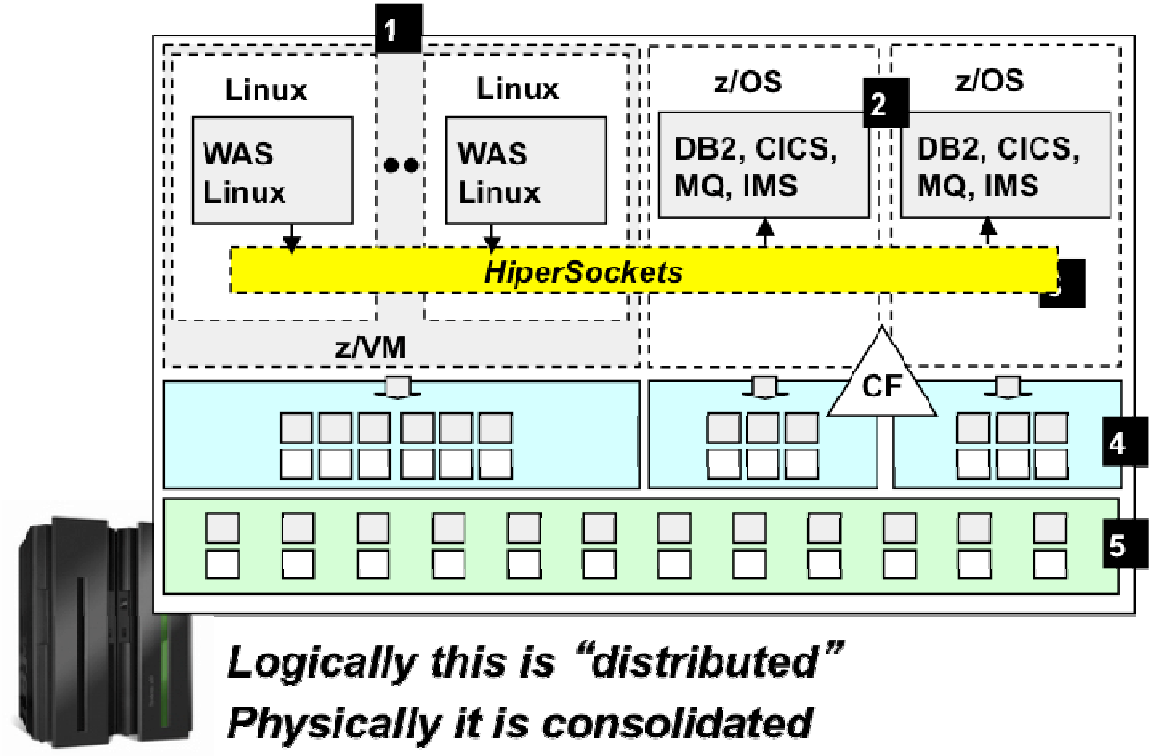
All the WAS z/OS pieces of that picture have been grayed-out

So too has the Parallel Sysplex and Coupling Facility

- **WAS for Linux on System z can not *directly* participate in Parallel Sysplex or the attributes of the z/OS operating system.**
- ***Active* exploitation of hardware virtualization through zVM is realized**
- ***Passive* benefit from the System z hardware is realized.**
- ***Indirect* participate in Parallel Sysplex is possible …**

Common Use of WAS Linux System z and Parallel Sysplex

Many access data on z/OS Parallel Sysplex from Linux LPARs in the same CEC, using HiperSockets for TCP access:



Logically this is "distributed"
Physically it is consolidated

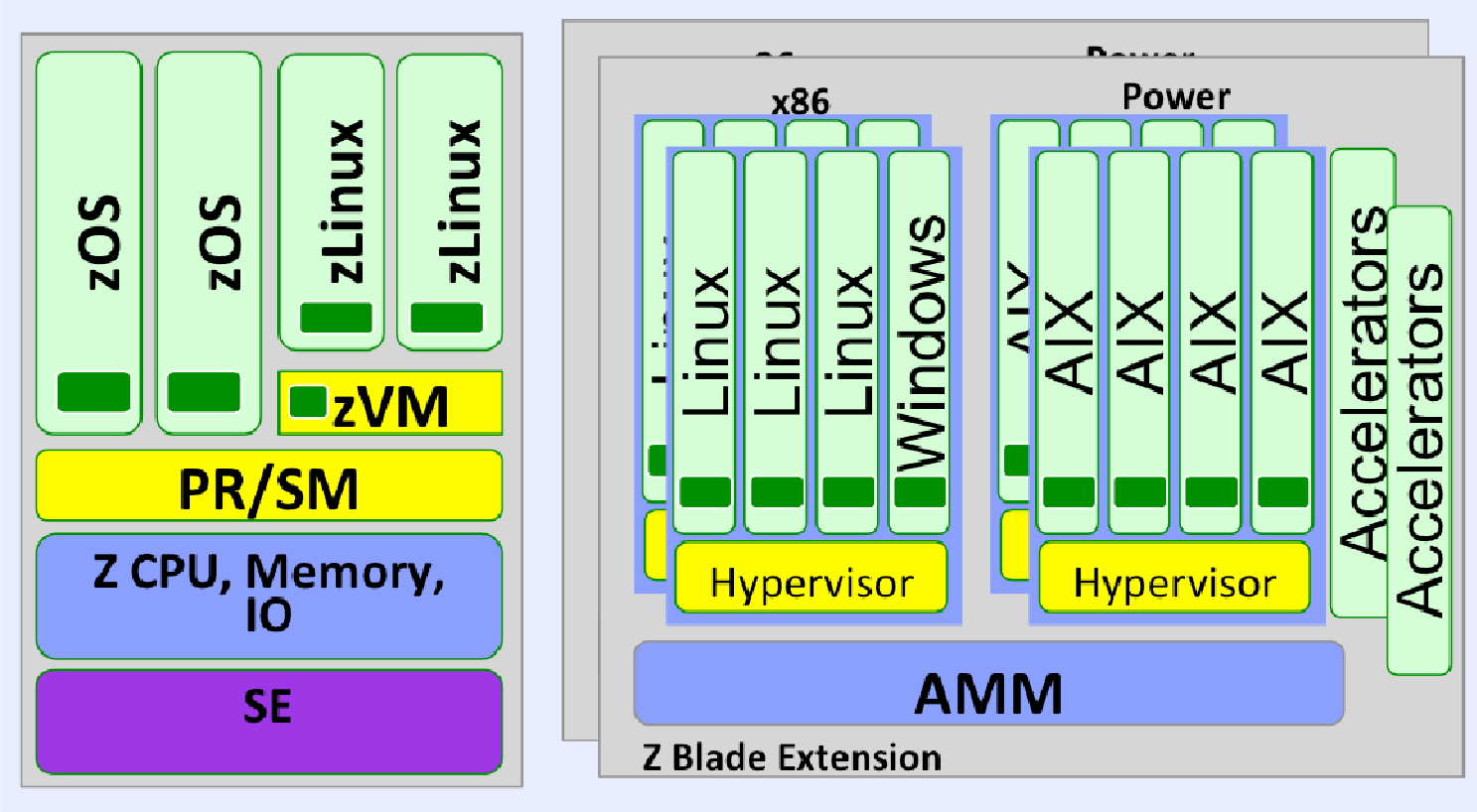
1. **Linux LPARs running WAS**
   - Exploiting virtualization of z/VM to realize server consolidation to single HW footprint
2. **Data subsystems in Parallel Sysplex**
   - Exploiting data sharing
3. **HiperSockets**
   - Optimized cross-LPAR virtual network
4. **Virtualized resources**
   - Exploiting LPAR technology of System z
5. **Real resources**
   - Efficient sharing of real resources through HiperVisor allocation of real to virtual

This is primarily used for server consolidation. Advantages over server farms include reduced power, cooling, square footage, administration and potentially software

Very good, but it does have drawbacks compared to co-location on z/OS …

WAS on Blades in the zBX

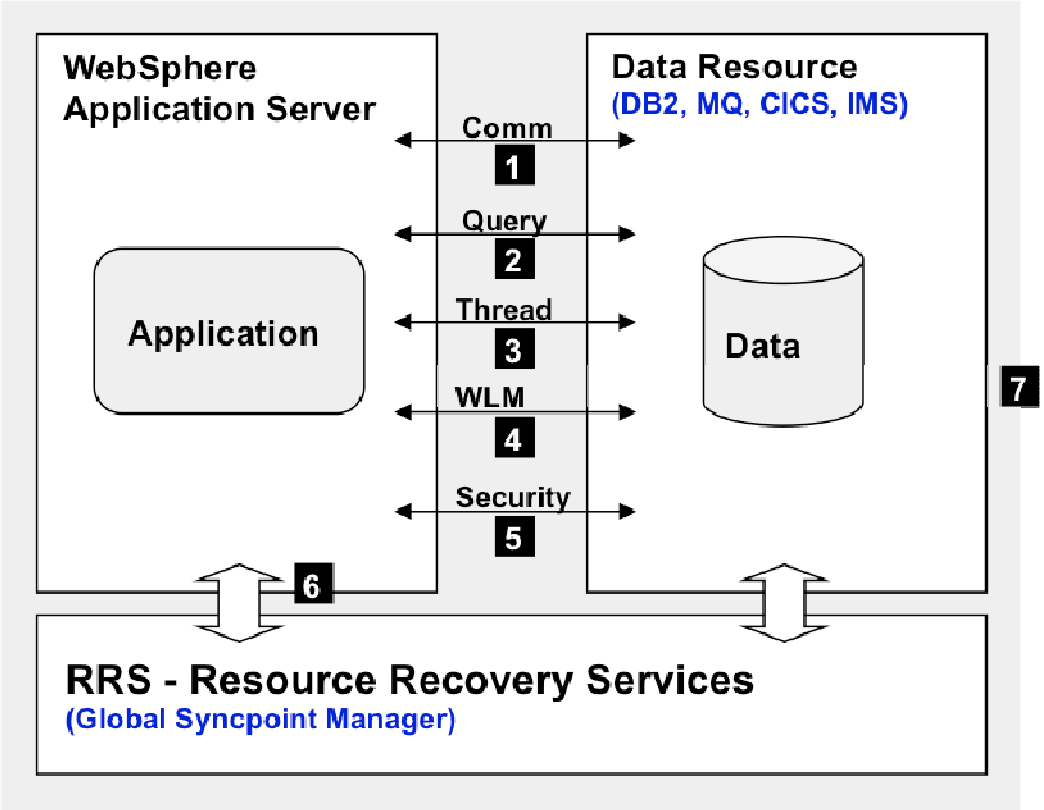
WebSphere servers can actively run on the blades within the zBX, and can be arranged into a cell that crosses the entire zEnterprise if needed



WAS running on xLinux/Windows/AIX can not *directly* participate in Parallel Sysplex or the attributes of the z/OS operating system.

- *Active* exploitation of hardware virtualization is realized
- Active Management of the blade hardware is realized
- *Passive* benefit from the System z hardware is realized.

Linux/Distributed vs. Co-Location on z/OS



1. **Cross-memory data transfer**
   - Better than even HiperSockets
2. **Avoid data and parameter serialization**
   - Since not passing across network, do not need to serialize. Avoids SSL as well.
3. **Single thread of execution**
   - Avoid switching threads, which means even greater efficiency
4. **Manage to a single WLM goal**
   - Easier goal definition and management
5. **Passing security context**
   - More options for security identity propagation: servant ID, client ID, application role vs. alias for remote T4
6. **Sysplex-wide RRS**
   - Extremely efficient transaction recovery processing
7. **Reduced complexity**
   - Single OS, more focused problem determination

- Important point here … solution architectures that span multiple operating systems environments implies different monitoring and management capabilities. Correlating information between those different tools can be challenging. Co-location helps reduce that complexity by bringing it all under a single operating system environment.

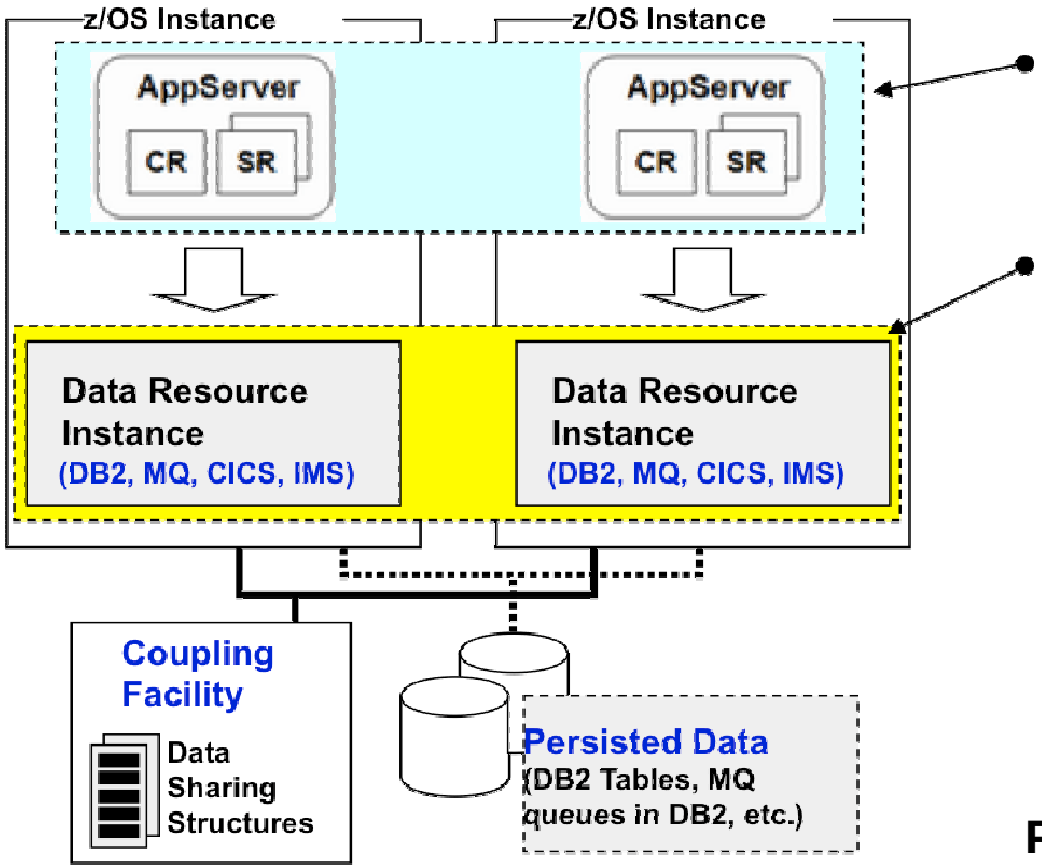
Availability and Scalability

Two of the more common business drivers for System z and z/OS

Impact2012
Comes To You

At The Heart -- Sysplex Data Sharing

Parallel Sysplex data sharing provides duplicated access to the same data. Data access and locking issues provided by Coupling Facility and Subsystems

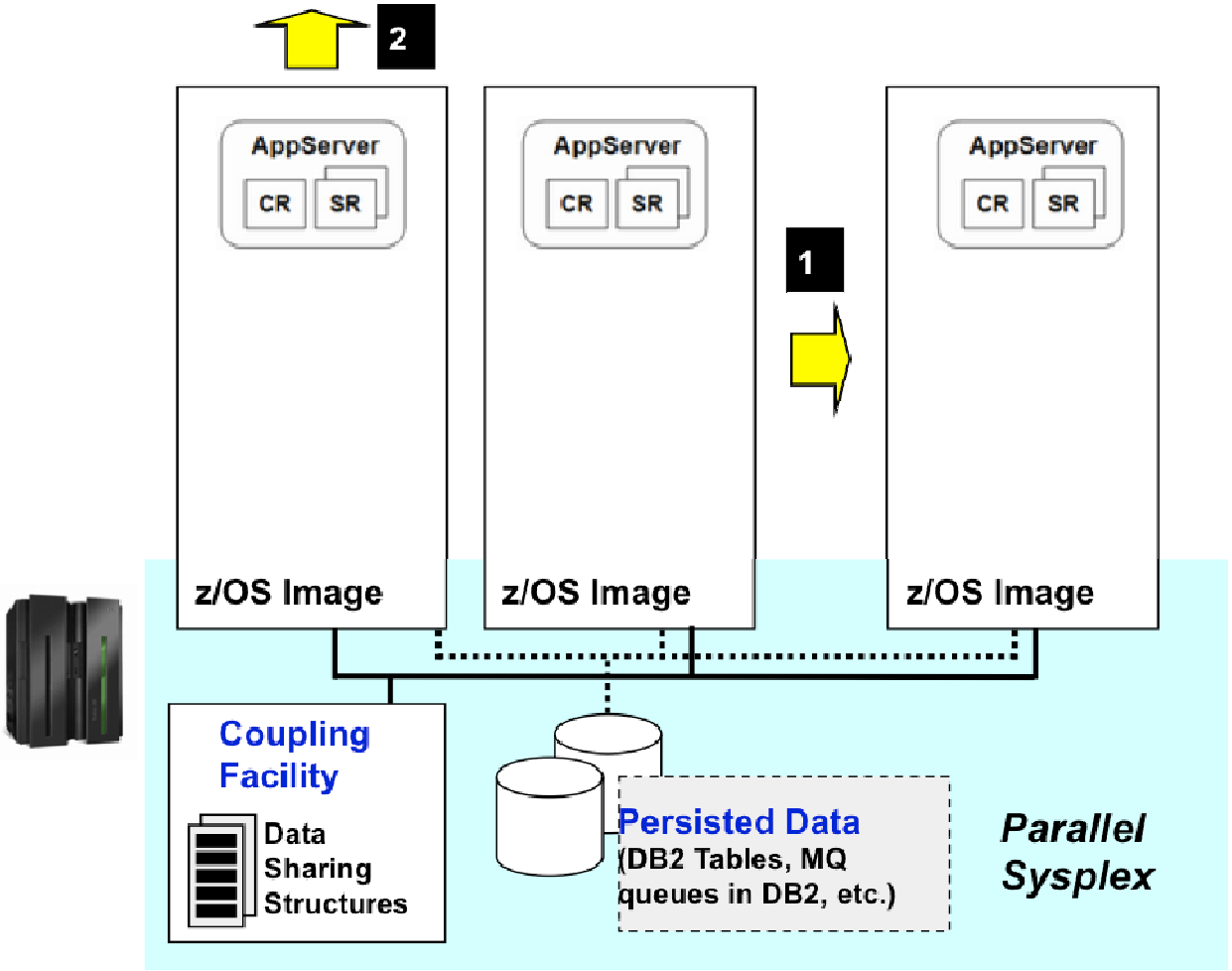


- **WebSphere "Cluster" consists of multiple physical application servers**
  - They are physically separate in most ways. Together it represents a logical one.
- **Sharing Group**
  - Physically separate instances but organized so they understand participants in group and have defined sharing relationship
- **AppServers and Resource Instances**
  - An application in an AppServer interacts with a data resource *instance*. Sharing conflicts are resolved by the data resource instances working in concert with each other with the help of z/OS and the Coupling Facility.

Parallel Sysplex and Data Sharing has been around for a decade and more. The technology is mature and proven and in use by large customers the world around.

Scalability

Two kinds of scalability -- Horizontal and Vertical



Horizontal Scaling

- This is what people most often think about when they think of scalability.
- It can work, but it gets increasingly difficult unless you have an effective shared resource (data) clustering mechanism.
- Parallel Sysplex is just such a mechanism.
- Shared disk storage systems, proven locking mechanisms, in-memory data structures and caching (CF) all make for effective horizontal scaling.

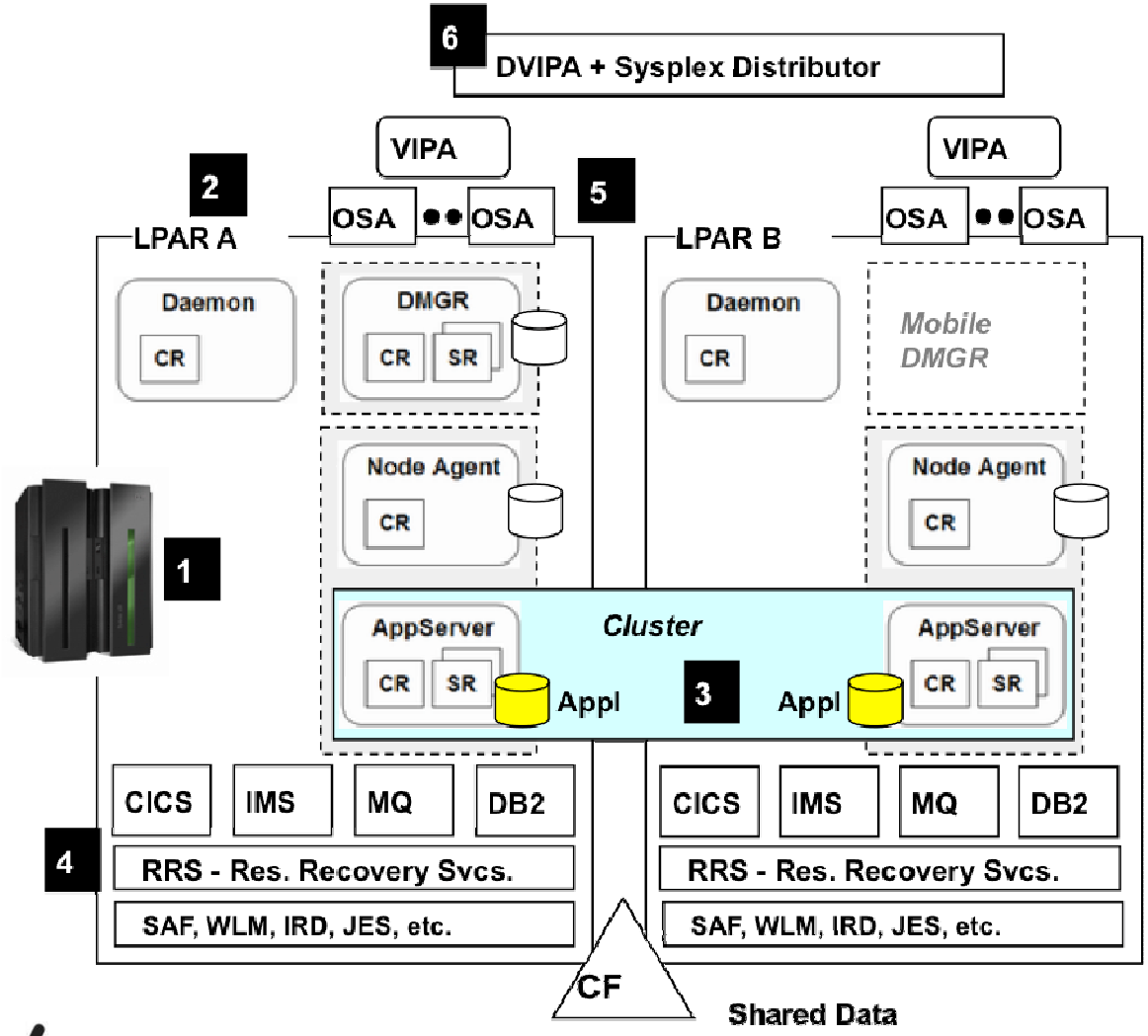
Vertical Scaling

- Vertical scaling is often overlooked. The result is massive horizontal scaling with all the attendant issues of manageability.
- z/OS is designed for high degrees of utilization and has the capability to scale very high per system image. The balanced architecture (CPU, memory, cache, I/O) allow for this.

System z and Parallel Sysplex provides both. That's the design point of the platform. That's how it's used in many large customer installations.

The Big Picture of WAS z/OS and Parallel Sysplex HA

It's all about redundancy *and* integration with platform HA function



1. **Redundant and fault-tolerant hardware**
   - System z hardware design has many layers of fault tolerance and redundancy.
2. **Redundant z/OS instances**
   - Either through logical partitioning (LPAR) or separate physical machines.
3. **Clustered WebSphere z/OS servers**
   - Multiple application servers grouped into a logical unit for application deployment and management
     - z/OS exclusive: dynamic SR expansion (more coming up)
4. **Redundant data resource managers with Sysplex shared data**
   - Multiple resource managers instances with shared data in CF and a global syncpoint manager (RRS)
5. **Redundant network adapters hidden behind Virtual IP address**
   - On the front end, multiple network interfaces with a moveable virtual IP address protecting against outage
6. **Workload distribution hidden behind distributed virtual IP and Sysplex Distributor**
   - Further abstraction of real IP addresses behind a virtual IP that can be swapped across images in a Sysplex, with Sysplex Distributor providing TCP connection distribution based on WLM

We love your Feedback!

- Don’t forget to submit your Impact session and speaker feedback! Your feedback is very important to us, we use it to improve our conference for you next year.
- Go to impactsmartsite.com from your mobile device
- From the Impact 2012 Online Conference Guide:
  - Select Agenda
  - Navigate to the session you want to give feedback on
  - Select the session or speaker feedback links
  - Submit your feedback

Impact2012
Comes To You

Copyright and Trademarks

Impact2012
Comes To You


Thank
You